NBSIR 83-2703

FASTMENU: A Set of FORTRAN Programs for Analyzing Surface Texture

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Center for Manufacturing Engineering
Washington, DC 20234

July 1983

U.S. DEPARTMENT OF COMMERCE

NATIONAL BUREAU OF STANDARDS

NBSIR 83-2703

FASTMENU: A SET OF FORTRAN PROGRAMS FOR ANALYZING SURFACE TEXTURE

T. V. Vorburger

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Center for Manufacturing Engineering
Washington, DC 20234

July 1983

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

A
d
m
p
2
a
r
c

ABSTRACT

A set of FORTRAN programs for surface texture analysis is described. These programs were developed for use with a minicomputer that is interfaced to stylus type instruments. The programs 1) perform data acquisition from the stylus instruments, 2) store the data on magnetic disk, and 3) perform statistical analyses for parameters such as the roughness average R_a, rms roughness R_q, and for the autocorrelation function and amplitude density function.

TABLE OF CONTENTS

1. Introduction

Surface texture needs are becoming better understood and more carefully specified for a wide range of industrial parts like ship hulls and propellers [1], automobiles [2], and x-ray mirrors [3]. Consequently, the measurement of surface texture is becoming more sophisticated as its importance for industrial products increases. More and more, the measurement of surfaces by stylus techniques involves digitization of the data and statistical time-series analysis both of which require a computer or microprocessor connected on line to the stylus instrument. Important components of these systems are the programs or software which direct the sequence of measurement operations.

At NBS, we have developed a system called FAST (Facility to Analyze Surface Texture) [4], which includes two commercial stylus instruments interfaced to a minicomputer that controls the data acquisition and analysis through a set of nine FORTRAN programs called FASTMENU. Each of these programs can be executed with a command from the computer console. Their names and functions are as follows:

ROUGHNES is used when measuring surface profiles for characterizing and calibrating surface roughness. As the stylus traverses the rough surface, the time varying electrical signal is digitized under program control and stored in the computer memory. The roughness average parameter R_a is calculated and printed for each profile, and the digitized profile data are stored on magnetic disks for subsequent analysis.

STEPHGHT, like ROUGHNES, controls data acquisition and storage, but this program is used for measurement and calibration of step heights. The step heights are calculated and printed during the execution of the program.

WHATSON is a utility program for examining the header information of data files stored on a disk. This program is helpful for keeping track of the various kinds of data and for determining which files may be deleted.

AVRGRA is used primarily in calibration reports. It calculates the average and the random uncertainty of a set of step height values or R_a values. The calibration uncertainty of the instrument and the resulting total uncertainty of the measurement are also calculated and all of the results are printed.

SMORGAS is a program that calculates seven parameters from stored profile data. These are R_a; the rms roughness - R_q; the peak-valley roughness - R_{tm}; the average slope - S_a; the average wavelength - D_a; the peak count wavelength D_{pc}; and the skewness - Q. The operator has the options of fitting the data to a least squares mean line, of choosing the sample length for R_{tm}, and of choosing the point to point spacing of the average slope calculation.

PLOTSVIL generates plots of the surface profiles stored on disk.

ADF, ACF and PSD are used to calculate and plot the amplitude density function, autocorrelation function, and power spectral density respectively for any or all the profiles in a file. These functions may also be printed as arrays of numbers if the operator chooses.

These programs were written in FORTRAN 77 for use on a Perkin Elmer-Interdata* 7/32 minicomputer with 256 Kbytes of memory storage, and magnetic disk storage consisting of 4 disks each with 5 Mbytes capacity. PSD is the largest by far of the nine programs. It requires 111 Kbytes of memory storage. The next largest module, ACF, requires 69 Kbytes.

The system also includes a system console, a printer, a Versatec printer-plotter, and an Analog-to-digital converter (ADC) housed in an interface (built at NBS) called the NBS Bus. The stylus instruments, a Talystep and a Talysurf 4, track the undulations of the surface profile and produce a time varying output voltage which can be filtered using one of several high pass electronic filters. A frequency generator interfaced to the NBS Bus controls the rate at which the output signal is digitized and hence the point-to-point spacing of the data.

The present arrangement is quite similar to one developed at NBS several years ago [5] with a much smaller minicomputer system. Whereas the previous software was written in machine code [6] because of the limited memory size, the present system size allows the use of the high level compiled language, FORTRAN 77. It should be noted that a few of the commands to be shown here are not part of the FORTRAN 77 language but come from run time libraries supplied by Perkin Elmer and Versatec.

One convenience of the present storage system is that each of the four disks serves a different function. The FORTRAN programs for surface texture measurement are stored on a removable disk called SRF, and the basic operating system programs are stored on the fixed, system disk, SYS. A second removable disk contains the surface profile data and, to date, approximately 12 disk cartridges have been filled with close to 60 Mbytes of profile information. The fourth disk is used as a scratch disk to hold the large temporary files which are needed for plotting graphs.

The software works as follows. Each of the nine commands listed above calls a file from the SYS disk. This file is a set of operating system commands that loads the appropriate program from

---

* Certain commercial equipment are identified in this report to specify adequately the experimental procedure. In no case does such identification imply recommendation or endorsement by the National Bureau of Standards, nor does it imply that the equipment identified is necessarily the best available for the purpose.

SRF, assigns various logical units and executes the program. After the program is completed, the operating system shows the menu again or refers the operator to it, so that the operator can be reminded of the nine commands that are available in the set.

The remainder of this report is organized as follows. Chapter 2 discusses the three operating system files that introduce the operator to the system. Chapters 3-11 describe the structure of the programs for carrying out the nine commands. Each chapter describes one command and consists of five parts:

1) a short summary of the program (what it does),
2) a copy of the operating system file which calls the program,
3) a copy of the FORTRAN program,
4) a flowchart of the FORTRAN program, and
5) a sample of the hardcopy output, i.e. printouts or plotted graphs.

During this software development project I was greatly aided by F. E. Scire, who thoroughly tested many aspects of the system and made a number of valuable suggestions, and by E. C. Teague, who wrote the data gathering subroutine ADCIO. I am also grateful to D. E. Gilsinn for several enlightening discussions, to S. A. Morris for preparation of the manuscript, to L. Greenspan, M. Cadoff, and B. Rust for reviewing it, and to the GAMS Support Services Group for providing the portable FFT programs from the Center for Applied Mathematics software library. The work was supported in part by the David Taylor Naval Ship Research and Development Center, Annapolis, Maryland.

2. The Menu

When the operator types the command "FAST", the text shown in fig. 2.1 appears on the console screen. The instructions point to two other commands, FASTMENU (fig. 2.2) and FASTNITE (fig. 2.3).

FASTMENU is the heart of the system. It lists the main programs in the package along with a short description for each, which reminds the operator about its function. FASTNITE is a supplementary set of instructions which explains how several of the analysis programs may be run in a batch mode so that the operator does not have to sit at the console and choose options. To do this correctly, the operator must know the sequence of responses that are needed for each program and type these ahead of time in a disk file. The text in FASTNITE simply explains how to instruct the program to look for the disk file for input parameters rather than the console.

```
*                    ***** FAST*****
*
*        WELCOME  TO THE WORLD OF SURFACE  TEXTURE  MEASUREMENT!
*
*
*        USE THE COMMAND  "FASTMENU"  TO FIND OUT ABOUT  PROFILE
*        MEASUREMENT,  ROUGHNESS  AND  STEP HEIGHT CALIBRATIONS,
*        AND THE ASSOCIATED ANALYSIS PROGRAMS.
*
*
*        USE THE COMMAND "FASTNITE"  TO FIND OUT HOW TO RUN SOME
*        OF THE ANALYSIS PROGRAMS IN A BATCH MODE,  FOR EXAMPLE,
*        OVERNIGHT.
```

Figure 2.1

```
*            ***** FASTMENU *****
*
* HERE  ARE  THE COMMANDS FOR MEASURING AND ANALYZING SURFACE PROFILE DATA.
*
* ROUGHNES  IS USED FOR MEASURING SURFACE ROUGHNESS, PARTICULARLY RA.
*
* STEPHGHT  IS USED FOR MEASURING STEP HEIGHTS.
*
* WHATSON VOL:   GIVES A RUNDOWN OF THE DATA FILES PRESENT ON A DISK (VOL:)
*           BY LISTING THE FIRST TEN LINES OF EACH FILE.
*
* AVRORA    CALCULATES THE AVERAGE FOR A SET OF RA  OR STEP HEIGHT DATA AND
*           THE VARIOUS UNCERTAINTIES AS WELL.
*
* SMORGAS   CALCULATES THE FOLLOWING SURFACE PARAMETERS:  RA, RQ, RTM, AVE.
*           SLOPE, AVE. WAVELENGTH, PEAK-COUNT WAVELENGTH, AND SKEWNESS.
*
* ACF       THESE COMMANDS  YIELD PLOTS  FOR  THE AUTOCORRELATION FUNCTION,
* ADF       AMPLITUDE DENSITY FUNCTION, AND POWER SPECTRAL DENSITY.
* PSD
*
* PLOTEVIL  YIELDS PLOTS OF SURFACE PROFILES.
*
* TO FIND OUT HOW TO USE "ACF", "ADF", AND "SMORGAS" IN AN OVERNIGHT, BATCH
* MODE, TYPE "FASTNITE".
```

Figure 2.2

```
*          ***** FASTNITE *****
*
* THE COMMANDS    "SMORGAS",  "PLOTSVIL",  "ACF",  AND "ADF"
* ALLOW  THE OPTION OF TYPING THE CONSOLE PARAMETERS  INTO A
* FILE AHEAD OF TIME. THIS ENABLES YOU TO ANALYZE DATA OVER-
* NIGHT WITHOUT SITTING BY THE SYSTEM CONSOLE.
*
* IN THE CASE OF "SMORGAS", THE COMMAND WOULD BE
*
* SMORGAS (FILENAME).
*
* LOGICAL UNIT  5  IS THEN ASSIGNED TO THE FILE SPECIFIED BY
* "FILENAME" RATHER THAN TO THE CONSOLE.
*
* "PLOTSVIL", "ACF", AND "ADF" WORK THE SAME WAY.
```

Figure 2.3

3. ROUGHNES

3.1 Summary

The program ROUGHNES controls the data acquisition of roughness profiles and calculates the roughness average R_a for each. The file of operating system commands (sec. 3.2), loads the program, assigns the logical units, and starts the execution. The program has a main part and four subroutines - KCAL1, RA1, STPHGT, ADCIO. The main program is used to create and assign the new file that will hold the profile data. The program also calls for header information to be entered into the file and used as a label during printouts. The main program then calls two subprograms, first KCAL1, then RA1.

KCAL1 is used to calculate a calibration constant KCAL for the apparatus. To do this, it calls the subroutine ADCIO, which controls the digitization of a surface profile of a calibration step [5] as the step is being measured in the stylus instrument. KCAL1 then calls STPHGT which fits straight lines to the profile and calculates a step in terms of quantization levels. Since the true height of the calibration step is known in micrometers (um), the subroutine is then able to calculate KCAL in units of um/quantization level. This calibration procedure is performed three times and the average KCAL is calculated and stored.

The subroutine KCAL1 also allows the option to read the calibration constant from a previously created file instead of recalibrating the instrument.

Control then passes to the RA1 subroutine which measures the surface profiles and calculates R_a using both the raw data and the calibration constant. To do this, RA1 calls ADCIO which digitizes the roughness profile. Each profile consists of 4000 digitized points. It is measured three times, and the R_a value is calculated for each time. Then an average of the three values is calculated and stored, and the third profile is stored on magnetic disk. The set of three profiles is called a position. For all of the data-taking and calibration runs, the operator has the option of rejecting the data profile if it seems incorrect or of beginning an entire position over again.

3.2 Operating System Commands

```
1     ***** ROUGHNES *****
2     *
3     LO .BG, SRF:ROUGHNES.TSK
4     T .BG
5     CLOSE ALL
6     AS 4,L4:
7     AS 5,L1:
8     AS 6,C:
9     XDE SRF:SPILL.DAT
10    AL SRF:SPILL.DAT,IN,80/5/5
11    AS 8,SRF:SPILL.DAT
12    $W* YOU SHOULD NOW BE RECEIVING INFORMATION ON THE SURFACE CONSOLE.
13    ST
14    *
15    * RETURN TO "FASTMENU"
16    *
17    FASTMENU
18    $EXIT
```

The above set of commands perform the following functions in the Perkin Elmer 7/32 computer.

LO loads the machine code program, SRF:ROUGHNES.TSK, into the background partition (.BG) of the memory.

T .BG is used to address the task in .BG for further information and direction.

CLOSE ALL closes all the logical units of the task, just in case any were still open.

The next three AS statements are used to assign logical units. L4: is the printer, L1: the surface console, and C: the system console.

XDE deletes the temporary print file, SRF:SPILL.DAT, which is left over from the previous running of ROUGHNES.

AL creates SRF:SPILL.DAT again. AS assigns it as logical unit 8. Note: The file that holds the profile data is created later on by SRF:ROUGHNES.TSK itself.

$W prints a message to the system console.

ST begins execution of the program.

* in lines 14-16 allows uninterpreted comments to be inserted into the command stream.

After the execution of the program is finished, FASTMENU (Fig. 2.2) is called and the main programs are listed on the system console again.

3.3 ROUGHNESS FORTRAN Program

```
C ***** SRF:ROUGHNES.FTN CALLED BY "ROUGHNES" *****
C
C
C THIS PROGRAM BEGINS THE SERIES FOR SURFACE ROUGHNESS
C MEASUREMENTS. THE MAIN PROGRAM ALLOWS LABELING DATA CON-
C CERNING THE DATE, THE SPECIMEN, AND THE INSTRUMENT TO BE READ
C INTO A FILE WITH THE FORMAT "VOL:ADATE.DAT". THIS FILE WILL
C EVENTUALLY CONTAIN ALL OF THE PROFILE DATA GENERATED DURING
C THE RUN AND IS CREATED BY THE OPERATOR IN THIS MAIN PROGRAM.
C       T. VORBURGER 4/13/78 (REVISED 9/82)
C
C
        INTEGER STAT1,STAT2,STAT,FDBUS(2)
        INTEGER*2 DATFIL(8),BUFFER(4000)
        COMMON /FUDGE/BUFFER
        DIMENSION SPEC(20),RINSTR(20)
        DATA DATFIL(7),DATFIL(8)/'.D','AT'/
        DATA FDBUS/'BUS:    '/
C
C
1       FORMAT(' ***** NBS COMPUTERIZED SURFACE ROUGHNESS FACILITY',
       1' IS READY TO GO. *****'///
       1' TYPE THE NAME OF THE DATA FILE FOR TODAY IN THE' /
       2' FOLLOWING FORMAT: "VOL:ADATE", WHERE VOL IS THE' /
       3' THREE CHARACTER VOLUME NAME AND ADATE IS THE EIGHT' /
       4' CHARACTER FILE NAME.' )
2       FORMAT(1H ,20A4)
3       FORMAT(' NBS COMPUTERIZED SURFACE ROUGHNESS FACILITY' )
4       FORMAT(' TYPE THE SPECIMEN ID INFORMATION.' )
5       FORMAT(20A4)
6       FORMAT(' I CAN"T OPEN THIS FILE. TRY AGAIN.')
7       FORMAT(1H ,9A2)
8       FORMAT(' TYPE "TALYSTEP" OR "TALYSURF" AND THE VERTICAL' /
       1' AND HORIZONTAL MAGNIFICATIONS.' )
9       FORMAT(' TYPE THE POINT-TO-POINT SPACING IN UM/PT.' /
       1' INCLUDE THE DECIMAL POINT EXPLICITLY.' )
10      FORMAT(F10.4)
11      FORMAT(' THE FIRST FOUR RECORDS IN ' ,8A2/
       1' ARE AS FOLLOWS:' )
12      FORMAT(8A2)
13      FORMAT(' THE POINT-TO-POINT SPACING IS ' ,F8.4,
       1'  UM.' )
17      FORMAT(' I CAN"T CREATE A FILE WITH THAT NAME.'/'COME ON TURKEY.',
       1' DO IT OVER AND GET IT RIGHT THIS TIME!' )
19      FORMAT(' NBS BUS ASSIGN ERROR. STATUS WAS:' ,2X,I4)
C
C
C FIRST, THE OPERATOR IS ASKED TO TYPE THE FILENAME, THAT WILL
C BE CREATED TO HOLD THE DATA, AND THE SAMPLE IDENTIFICATION.
C
C
92      WRITE(5,1)
        READ(5,12) (DATFIL(J),J=1,6)
        CALL CFILW(DATFIL,2,80,5,5,0,0,STAT1)
        IF (STAT1.LT.1) GO TO 90
        WRITE(5,17) STAT1
        GO TO 92
90      CALL OPENW(1,DATFIL,4,0,0,STAT2)
```

```
      IF (STAT2 .LT. 1) GO TO 93
      WRITE(5,6)
      GO TO 92
C
C
C NOW, THE OPERATOR IS ASKED TO TYPE THE HEADER INFOR-
C MATION CONCERNING THE SPECIMEN, THE INSTRUMENT, AND
C THE SPACING OF THE DATA POINTS.
C
C
93    WRITE(5,4)
      READ(5,5) SPEC
      WRITE(5,8)
      READ(5,5) RINSTR
      WRITE(5,9)
      READ(5,10) PTTOPT
      WRITE(5,11) DATFIL
      WRITE(5,7) DATFIL
      WRITE(5,2) SPEC
      WRITE(5,2) RINSTR
      WRITE(5,10) PTTOPT
      WRITE(1,12) DATFIL
      WRITE(1,5) SPEC
      WRITE(1,5) RINSTR
      WRITE(1,10) PTTOPT
      WRITE(8,3)
      WRITE(8,7) DATFIL
      WRITE(8,2) SPEC
      WRITE(8,2) RINSTR
      WRITE(8,13) PTTOPT
C
C
C AT THIS POINT WE OPEN THE NBS BUS AS LU-3.
C
C
50    CALL OPENW(3,FDBUS,4,0,0,STAT)
      IF (STAT .EQ. 0) GO TO 105
      WRITE (5,19) STAT
      PAUSE 1
      GO TO 50
C
C
C FINALLY, THIS PROGRAM CALLS THE KCAL1 AND RA1
C SUBROUTINES ONE AT A TIME.
C
C
105   CALL KCAL1(DATFIL)
      CALL RA1(DATFIL)
      CLOSE(1)
      CLOSE(3)
      CLOSE(8)
      STOP
      END
C
C
C
C
C
```

```
      SUBROUTINE KCAL1(DATFIL)
C
C
C THIS ROUTINE CALCULATES THE CALIBRATION CONSTANT FOR THE
C SURFACE ROUGHNESS FACILITY. THERE ARE TWO OPTIONS.
C THE OPERATOR MAY USE THE INFORMATION AND CALIBRATION
C CONSTANTS ALREADY STORED IN A PREVIOUS "VOL:ADATE.DAT"
C FILE OR MAY CHOOSE TO RECALIBRATE THE INSTRUMENT.
C THE RESULTS ARE RECORDED IN THE CURRENT "VOL:ADATE.DAT" FILE.
C        T. VORBURGER   4/17/78   (REVISED 9/82)
C
C
      INTEGER*2 DATFIL(8),Q1,Q2,OLDFIL(8)
      DIMENSION SPEC(20),RINSTR(20),TALY(2)
      DIMENSION NKCAL(3),IE1(3),IE2(3),H1(3),H3(3),H5(3)
      DIMENSION H7(3),H4(3)
      REAL KCAL(3)
      INTEGER STAT2,FDBUS(2)
      INTEGER*2 BUFFER(4000)
      COMMON /FUDGE/BUFFER
      DATA IBLANK/'    '/
      DATA FDBUS/'BUS:    '/
      DATA OLDFIL(7),OLDFIL(8)/'.D','AT'/
C
1     FORMAT(' WE NOW BEGIN THE RECALIBRATION PROCEDURE.')
2     FORMAT(1H /' ADJUST THE FREQUENCY GENERATOR TIMING FOR'/
     1' STEP MEASUREMENT ON THE *** ',2A4,' ***.'/
     1' AND SET THE FILTER FOR "STEPS" MODE.')
3     FORMAT(40A2)
4     FORMAT(' BAD DATA? WE"LL START THE STEP CALIBRATION OVER.')
5     FORMAT(20A4)
6     FORMAT(' WHAT IS THE HEIGHT OF THE CALIBRATING STEP IN UM?'/
     2' INCLUDE THE DECIMAL POINT.')
7     FORMAT(F10.4)
8     FORMAT(' CALIBRATING STEP = ',F10.4,' UM'/////)
9     FORMAT(4X,'H1',8X,'H3',8X,'H5',8X,'H7'/4F10.4)
10    FORMAT(' STEP HEIGHT = ',F10.4,' UM')
11    FORMAT(' KCAL = ',E13.6,' UM/QUANTIZATION LEVEL')
12    FORMAT(' LOOK OK? TYPE "YES", "NO", OR "ST"(FOR START OVER).')
13    FORMAT(3(/10X,'TRACE',I3/10X,'THE EXTREMA ARE',I3,' AND ',
     1I3,' MM.'/10X,4X,'H1',8X,'H3',8X,'H5',8X,'H7'/10X,4F10.4/10X,
     2'STEP HEIGHT = ',F10.4,' UM'/10X,'KCAL = ',E13.6,
     3' UM/QUANTIZATION LEVEL'))
14    FORMAT(E13.6)
15    FORMAT(8A2)
16    FORMAT(//' THE FIRST SIX RECORDS IN TODAY"S DATA FILE ARE:'/
     11X,8A2/1X,20A4/1X,20A4/F10.4,' UM/PT(SPACING)'/F10.4,' UM =',
     2' THE HEIGHT OF THE CALIBRATING STEP.'/E13.6,
     3' UM/QUANTIZATION LEVEL = KCAL')
17    FORMAT(20A4/20A4/F10.4/F10.4/E13.6)
18    FORMAT(' DO YOU WISH TO RECALIBRATE THE INSTRUMENT?'/
     1' "Y" OR "N"?')
19    FORMAT(' ***** RESET STYLUS INSTRUMENT.'/
     1' ***** THEN  HIT RETURN KEY, WHEN YOU ARE READY TO TAKE DATA.')
22    FORMAT(' DATA READING COMPLETE.'/1X,I4,2X,' POINTS',
     1' OVERFLOW.'/1X,I4,2X,' POINTS UNDERFLOW.')
29    FORMAT(' WHAT IS THE NAME OF THE FILE THAT YOU',
     1' WANT THE KCAL FROM?'/' USE THE TWELVE CHARACTER',
```

```
     2' FORMAT "VOL:ADATE",'/' WHERE "VOL" IS THE THREE',
     3' CHARACTER VOLUME NAME'/' AND "ADATE" IS THE',
     4' EIGHT CHARACTER FILE NAME.')
30    FORMAT(' ERROR CODE',I3/' SOMETHING"S WRONG.',
     1' I CAN"T OPEN THIS FILE.'/' TRY AGAIN.')
32    FORMAT(/////' THE CALIBRATION CONSTANT KCAL HAS ',
     1'BEEN OBTAINED FROM THE FILE ',9A2)
C
C
C IN THE NEXT SEQUENCE, THE OPERATOR IS ASKED TO DECIDE
C WHETHER TO RECALIBRATE THE INSTRUMENT OR TO USE A
C CALIBRATION CONSTANT FROM ANOTHER FILE.
C
C
122   WRITE(5,18)
      READ(5,3) Q1
      IF(Q1.EQ.'N') GO TO 120
      GO TO 126
120   WRITE(5,29)
      READ (5,3) (OLDFIL(J),J=1,6)
      CALL OPENW (2,OLDFIL,0,0,0,STAT2)
      IF (STAT2 .LT. 1) GO TO 121
      WRITE (5,30) STAT2
      GO TO 122
121   READ (2,7,REC=5) CALSTP
      READ (2,14) AVKCAL
      WRITE (8,32) (OLDFIL(J),J=1,6)
      WRITE(8,8) CALSTP
      CALL CLOSE (2,STAT2)
      WRITE (1,7,REC=5) CALSTP
      WRITE (1,14) AVKCAL
      GO TO 127
C
C
126   WRITE(5,1)
      READ(1,5,REC=3)TALY
      WRITE(5,2) TALY
      IF(TALY(2).EQ.'SURF') GO TO 104
      HORSPC = 10.
      GO TO 105
104   HORSPC = 6.6667
C
C
C KCAL ROUTINE: THE DATA ARE READ INTO THE COMPUTER VIA
C THE ADCIO ROUTINE. THE RESULTING UNCALIBRATED STEP
C PROFILE IS FITTED BY TWO STRAIGHT LINES  IN THE STEP
C FITTING SUBROUTINE,STEPHGT. THE RESULT IS PASSED BACK TO
C THE MAIN PROGRAM AS THE VARIABLE Y4, WHICH IS DIVIDED
C THE THE HEIGHT OF THE CALIBRATING STEP TO YIELD A VALUE
C FOR KCAL IN UM/QUANTIZATION LEVEL. THE AVERAGE VALUE OF KCAL
C IS STORED IN AVKCAL AND WRITTEN INTO THE FILE "VOL:ADATE.DAT".
C
C
105   WRITE(5,6)
      READ(5,7) CALSTP
      WRITE(1,7,REC=5) CALSTP
C
106     SUM1 = 0.
```

```
                DO 102 K=1,3
                NKCAL(K) = K
C
C
C AT THIS POINT WE REMIND THE OPERATOR TO RESET THE INSTRUMENT.
C
103             WRITE (5,19)
                READ (5,5) JUNK
C
C
                CALL ADCIO(1000,Y'2307',4,3,80)
C
C
C INFORM OPERATOR OF COMPLETION OF DATA READING AND AMOUNT
C OF UNDERFLOW AND OVERFLOW.
C
                L1 = 0
                L2 = 0
                DO 75 I = 1, 1000
                IF (BUFFER(I) .EQ. 2047) L1 = L1 + 1
                IF (BUFFER(I) .EQ. -2048) L2 = L2 + 1
75              CONTINUE
                WRITE (5, 22) L1, L2
C
C
                CALL STPHGT(Y1,Y3,Y4,Y5,Y7,IE1(K),IE2(K),HORSPC)
                KCAL(K) = CALSTP/Y4
                H1(K) = KCAL(K)*Y1
                H3(K) = KCAL(K)*Y3
                H4(K) = KCAL(K)*Y4
                H5(K) = KCAL(K)*Y5
                H7(K) = KCAL(K)*Y7
                WRITE(5,9) H1(K),H3(K),H5(K),H7(K)
                WRITE(5,10) H4(K)
                WRITE(5,11) KCAL(K)
                WRITE(5,12)
                READ(5,3) Q2
                IF( Q2.EQ.'NO') GO TO 103
                IF(Q2.NE.'ST') GO TO 102
                WRITE(5,4)
                GO TO 106
102             SUM1 = SUM1+KCAL(K)
            AVKCAL = SUM1/3.
            WRITE(1,14) AVKCAL
            WRITE(8,13) (NKCAL(K),IE1(K),IE2(K),H1(K),H3(K),
           1H5(K),H7(K),H4(K),KCAL(K),K=1,3)
C
C
C THE LABELING INFORMATION CONTAINED IN THE FILE IS
C READ BACK AND PRINTED OUT FOR THE OPERATOR TO CHECK.
C THIS INCLUDES THE FILE NAME, THE SPECIMEN AND INSTRUMENT
C INFORMATION, THE POINT-TO-POINT SPACING, THE CALIBRATING
C STEP HEIGHT, AND KCAL(K).
C
C
127         READ(1,15,REC=1)DATFIL
            READ(1,17)SPEC,RINSTR,PTTOPT,CALSTP,AVKCAL
            WRITE(5,16) DATFIL,SPEC,RINSTR,PTTOPT,CALSTP,AVKCAL
```

```
          WRITE(8,16) DATFIL,SPEC,RINSTR,PTTOPT,CALSTP,AVKCAL
          RETURN
          END
C
C
C
C
          SUBROUTINE STPHGT(Z1,Z3,Z4,Z5,Z7,E1,E2,HORSPC)
C
C
C STEP FITTING ROUTINE: THIS ROUTINE FITS A STRAIGHT
C LINE TO THE SAMPLE PROFILE ON EACH SIDE OF THE STEP
C WHICH IS RECORDED IN THE ARRAY BUFFER/STPDAT. THE
C OPERATOR MUST CHOOSE EXTREMA, E1 AND E2, WHICH ARE
C SYMMETRICAL ABOUT THE STEP LOCATION. THE RESULT IS
C THE STEP HEIGHT IN QUANTIZATION LEVELS WHICH IS
C STORED IN THE VARIABLE Z4.  Z1,3,5,7 STORE THE HEIGHT
C DIFFERENCE BETWEEN THE LINES AT OTHER PLACES THAN AT
C THE STEP LOCATION.  HORSPC IS A CONVERSION FACTOR WHICH
C RELATES POSITION ON THE CHART RECORD IN MM TO POSITION
C IN THE DATA ARRAY,BUFFER.
C
          INTEGER*2 BUFFER(1000)
          COMMON /FUDGE/BUFFER
          INTEGER E1,E2,E3,E4,X1,X2
          DIMENSION Y1(250),Y2(250)
1         FORMAT(' TYPE EXTREMUM FOR LEFT HAND LINE IN MM'/
         1' USE I3 FORMAT.')
2         FORMAT(I3)
3         FORMAT(' TYPE EXTREMUM FOR RIGHT HAND LINE IN MM.'/
         1' USE I3 FORMAT.')
4         FORMAT(' YOU CAN TYPE BETTER THAN THAT. TRY AGAIN.'/)
51        WRITE(5,1)
          READ(5,2) E1
          WRITE(5,3)
          READ(5,2) E2
          E3 = HORSPC*E1
          E4 = HORSPC*E2
          X1 = E3-250
          X2 = E4-1
          IF (X1 .LT. 5) GO TO 53
          IF (X2 .GT. 751) GO TO 53
          GO TO 52
53        WRITE(5,4)
          GO TO 51
C
52          DO 100 J=1,250
            J1 = X1+J
            Y1(J) = BUFFER(J1)
            J2 = X2+J
100         Y2(J) = BUFFER(J2)
C
            S1 = 0.
            S2 = 0.
            T1 = 0.
            T2 = 0.
            U1 = 0.
            U2 = 0.
```

```
             V1 = 0.
             V2 = 0.
             DO 101 J=1,250
             S1 = S1+(X1+J)**2
             S2 = S2+(X2+J)**2
             T1 = T1+Y1(J)
             T2 = T2+Y2(J)
             U1 = U1+(X1+J)
             U2 = U2+(X2+J)
             V1 = V1+(X1+J)*Y1(J)
  101        V2 = V2+(X2+J)*Y2(J)
C
        D3 = 250.*S1 - U1**2
        D4 = 250.*S2 - U2**2
        RM1 = (250.*V1-U1*T1)/D3
        RM2 = (250.*V2-U2*T2)/D4
        B1 = (S1*T1-U1*V1)/D3
        B2 = (S2*T2-U2*V2)/D4
        B3 = B2-B1
        O1 = 0.125*E4+.875*E3
        O3 = 0.375*E4+.625*E3
        O4 = 0.5*(E4+E3)
        O5 = 0.625*E4+0.375*E3
        O7 = 0.875*E4+0.125*E3
        Z1 = (RM2-RM1)*O1+B3
        Z3 = (RM2-RM1)*O3+B3
        Z4 = (RM2-RM1)*O4+B3
        Z5 = (RM2-RM1)*O5+B3
        Z7 = (RM2-RM1)*O7+B3
        RETURN
        END
C
C
C
C
        SUBROUTINE RA1(DATFIL)
C
C
C THIS PROGRAM RECORDS THE AVERAGE ROUGHNESS RA FOR A 4000
C POINT SURFACE PROFILE. THE DATA ARE READ INTO THE ARRAY
C BUFFER VIA THE SUBROUTINE ADCIO. THE AVERAGE ROUGHNESS
C FOR EACH OF THREE TRACES AT EACH POSITION IS CALCULATED
C AND WRITTEN ONTO THE SURFACE CONSOLE AND THE PRINTER.
C THE AVERAGE OF RA'S FOR EACH POSITION IS WRITTEN
C ON THE FILE "SRF:ADATE.DAT". THE LAST PROFILE FOR EACH
C POSITION IS ALSO RECORDED IN "SRF:ADATE.DAT" ALONG WITH
C THE TOTAL NUMBER OF POSITIONS MEASURED FOR THE DAY.
C
C                                                    TVTV 4/22/78
C
C
        INTEGER*2 BUFFER(4000),DATFIL(8),NTRACE(3),NPOS,QUES1,
       1QUES2
        INTEGER SUM1,COMENT(10)
        DIMENSION RA(3),REGURG(20)
        REAL MEAN,KCAL
        COMMON /FUDGE/BUFFER
C
```

```
C
1      FORMAT(//' SURFACE ROUGHNESS RA MEASUREMENTS' /
     1' ***** ADJUST THE TIMING AND SELECT THE APPROPRIATE FILTER.')
2      FORMAT(40A2)
3      FORMAT(E13.6)
4      FORMAT(' POSITION',I3,'   TRACE',I3,'   RA =',F10.4,' UM')
5      FORMAT(' LOOK OK? "YES","NO", OR "ST"(START POSITION OVER).')
6      FORMAT(16I5)
7      FORMAT(/' POS. ' ,I2,'   AVERAGE RA =' ,F9.4,' UM; ' ,
     110A4,3(/' TRACE ' ,I3,'     RA =' ,F10.4,'  UM' ))
8      FORMAT(' DO ANOTHER POSITION?' )
9      FORMAT(' BAD DATA. YOU CHOOSE TO START THE POSITION' ,
     1'  OVER.' )
10     FORMAT(I2,F10.4,10A4)
11     FORMAT(' RA ROUTINE COMPLETED.' )
12     FORMAT(20A4)
13     FORMAT(///' SURFACE ROUGHNESS RA MEASUREMENTS' )
14     FORMAT(1H1)
15     FORMAT(1H ,40A2)
16     FORMAT(1H ,20A4)
17     FORMAT(' DO YOU HAVE ANY COMMENTS FOR THIS POSITION?'/
     1' YOU ARE ALLOWED 40 CHARACTERS.'/
     2' IF NOT, JUST HIT RETURN.')
18     FORMAT(10A4)
19     FORMAT(' ***** RESET STYLUS INSTRUMENT.'/
     1' ***** THEN  HIT RETURN KEY, WHEN YOU ARE READY TO TAKE DATA.')
22     FORMAT(' DATA READING COMPLETE.'/1X,I4,2X,' POINTS',
     1' OVERFLOW.'/1X,I4,2X,' POINTS UNDERFLOW.')
C
C
C FIRST, THE PROGRAM OPENS THE DATA FILE FOR THE DAY. ITS NAME
C HAS THE FORMAT "VOL:ADATE.DAT", AND IT IS STORED IN THE
C VARIABLE DATFIL. THEN THE VALUE OF THE CONSTANT KCAL
C IS READ FROM THE DATA FILE. THEN THE VARIABLE NPOS,
C GIVING THE NUMBER OF POSITIONS MEASURED, IS INDEXED AND
C RECORDED IN THE DATA FILE FOR THE DAY.
C
C
       WRITE(5,1)
       WRITE(8,13)
       READ(1,3,REC=6) KCAL
       NPOS = 1
       WRITE(1,15) NPOS
C
C NOW, WE BEGIN THE PROCESS OF READING PROFILE DATA AND
C CALCULATING RA. THIS IS DONE THREE TIMES FOR EACH
C POSITION. THE OPERATOR HAS THE OPTION OF ACCEPTING OR
C REJECTING EACH RA OR OF BEGINNING THE POSITION OVER.
C
105    SUM3 = 0.
         DO 100 K=1,3
         NTRACE(K) =  K
C
C
C AT THIS POINT WE REMIND THE OPERATOR TO RESET THE INSTRUMENT.
C
 103     WRITE (5,19)
         READ (5,12) JUNK
```

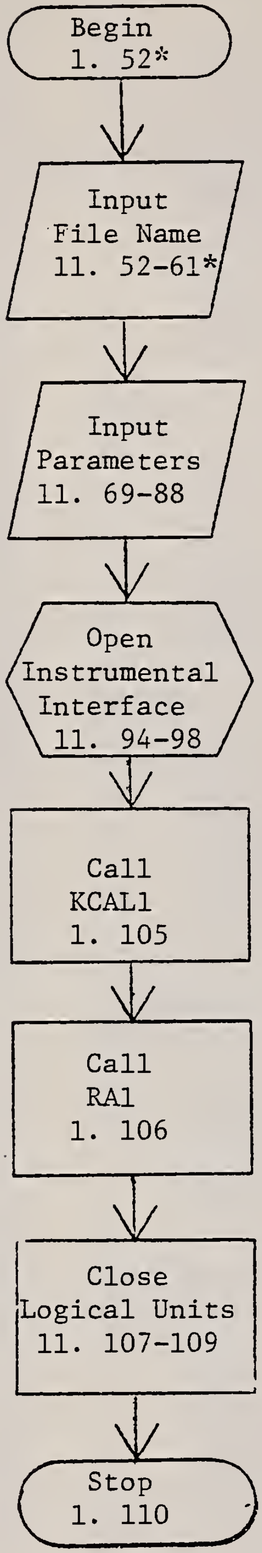
```
C
C
            CALL ADCIO(4000,Y'2307',4,3,40)
C
C
C INFORM OPERATOR OF COMPLETION OF DATA READING AND AMOUNT
C OF UNDERFLOW AND OVERFLOW.
C
            L1 = 0
            L2 = 0
            DO 75 I = 1, 4000
            IF (BUFFER(I) .EQ. 2047) L1 = L1 + 1
            IF (BUFFER(I) .EQ. -2048) L2 = L2 + 1
75          CONTINUE
            WRITE (5, 22) L1, L2
C
C
            SUM1 = 0.
              DO 101 J=1,4000
101           SUM1 = SUM1 + BUFFER(J)
            MEAN = SUM1/4000.
            SUM2 = 0.
              DO 102 J=1,4000
              DIFF = BUFFER(J) - MEAN
102           SUM2 = SUM2 + ABS(DIFF)
            RA(K) = KCAL*SUM2/4000.
            WRITE(5,4) NPOS,NTRACE(K),RA(K)
            WRITE(5,5)
            READ(5,2) QUES1
            IF (QUES1.EQ.'NO') GO TO 103
            IF (QUES1.NE.'ST') GO TO 100
            WRITE(5,9)
            GO TO 105
100         SUM3 = SUM3 + RA(K)
C
C THE AVERAGE IS CALCULATED, AND THE RESULTS ARE WRITTEN
C ON THE SURFACE CONSOLE, THE PRINTER, AND THE DATA FILE
C FOR THE DAY. THE PROFILE DATA FROM TRACE 3 FOR EACH
C POSITION ARE ALSO WRITTEN INTO THE DATA FILE. THE OPER-
C ATOR HAS THE OPTION OF GOING TO A NEW POSITION OR OF
C ENDING THE PROGRAM.
C
         AVRA = SUM3/3.
         WRITE(5,17)
         READ(5,18) COMENT
         WRITE(1,10) NPOS,AVRA,COMENT
         WRITE(1,6) BUFFER
         IF (NPOS.EQ.6 .OR. NPOS.EQ.16  .OR. NPOS.EQ.26
        1 .OR. NPOS.EQ.36 .OR. NPOS.EQ.46 .OR. NPOS.EQ.56
        2 .OR. NPOS.EQ.66 .OR. NPOS.EQ.76 .OR. NPOS.EQ.86)
        3 WRITE(8,14)
         WRITE(8,7) NPOS,AVRA,COMENT,(NTRACE(K),RA(K),K=1,3)
         WRITE(5,8)
         READ(5,2) QUES2
         IF(QUES2.EQ.'NO') GO TO 104
         NPOS = NPOS +  1
         GO TO 105
104      WRITE(1,10,REC=7) NPOS
```

```
      REWIND 8
      WRITE(4,14)
106   READ(8,12,END=108) REGURG
      IF (REGURG(1) .NE. '1   ') GO TO 107
      WRITE(4,14)
      GO TO 106
107   WRITE(4,16) REGURG
      GO TO 106
108   WRITE(5,11)
      RETURN
      END
C
C
C
C
C
      SUBROUTINE ADCIO (N, MDEADR, ADR, LU, TOC)
C
C THIS ROUTINE SETS UP THE NBS BUS FOR ACQUIRING A RECORD OF N DATA
C POINTS VIA THE A TO D CONVERTER (ADC). THE SETUP OF THE ADC
C ASSUMES THAT IT WILL BE EXTERNALLY CLOCKED AT THE REQUIRED
C RATE.  THE PROGRAM IS BASED ON THE BUS DRIVER WRITTEN BY
C CHARLES CODLING OF COMPUTER CONSULTANTS WHICH WAS IN TURN
C A MODIFICATION OF THE MUXBUS DRIVER WRITTEN BY RICHARD
C FREEMIRE OF NBS.
C
C THE PROGRAM ALERTS THE OPERATOR OF ANY PROBLEM IN OPENING
C THE BUS BY PROMPTS ON THE USER'S TERMINAL WHICH IS
C ASSUMED TO BE ASSIGNED TO LOGICAL UNIT 5.
C
C                    WRITTEN BY E. CLAYTON TEAGUE 4/10/78
C                    REVISED 1/14/83
C
C -----------------------------------------------------------
C INPUT VARIABLE DEFINITIONS.
C
C N - NUMBER OF DATA POINTS TO BE ACQUIRED. 2N BYTES WILL
C     BE ACQUIRED.
C MDEADR- HEX CODE TO INDICATE WHICH CHANNEL OF THE ADC
C         WILL BE USED IN THE DATA ACQUISITION, WHAT TYPE
C         OF CONVERSION TO MAKE ETC.  THE USER SHOULD CONSULT
C         A MANUAL ON THE NBS BUS TO DETERMINE WHAT THIS
C         CODE SHOULD BE.
C ADR - SUBADDRESS OF THE ADC ON THE NBS BUS.
C LU - LOGICAL UNIT ASSIGNMENT OF THE NBS BUS.
C BUFFER - STARTING ADDRESS OF THE STORAGE AREA FOR
C          ACQUIRED DATA.  IT IS THE USER'S RESPONSIBILITY
C          TO INSURE THAT 2N BYTES OF SPACE ARE ALLOTED.
C TOC - TIME OUT CONSTANT TO BE USED FOR THE DATA ACQUISITON.
C
C
      INTEGER DUMMY, ADR, TOC
      INTEGER PBLK(8), ISTAT
      INTEGER*2 BUFFER(4000)
      LOGICAL FLAG
      COMMON /FUDGE/BUFFER
      DATA PBLK/8*0/
C
```

```
C SET UP BUS FOR I/O  BY INITIALIZING BUSY BIT.
C
 30   FLAG = .FALSE.
 40   PBLK(6) = 0
C
C CLEAR BUS AND CHECK STATUS
C
      CALL SYSIO (PBLK, Y'C0', LU, DUMMY, 0, Y'C0')
      CALL SYSIO (PBLK, Y'C0', LU, DUMMY, 0, Y'B0')
      CALL ILBYTE(ISTAT, PBLK, 2)
      IF (ISTAT.EQ.2) GO TO 50
      WRITE (5, 930) ISTAT
      PAUSE 10
      GO TO 40
C
C SELECT ADC SUBADDRESS ON BUS AND CHECK STATUS.
C
 50   IADR = ADR + Y'C0'
      CALL SYSIO(PBLK, Y'C0', LU, DUMMY, 0, IADR)
C
      CALL ILBYTE (ISTAT, PBLK, 2)
      IF (ISTAT.EQ.136.OR.ISTAT.EQ.0)GO TO 60
      WRITE (5, 940) ISTAT
      PAUSE 11
      GO TO 50
C
C SET UP MODE AND ADDRESS REGISTER OF ADC.
C
 60   IMDE = MDEADR * Y'10000'
      CALL SYSIO(PBLK, Y'38', LU, IMDE, 2, 0)
      CALL ILBYTE(ISTAT,PBLK,2)
      IF(ISTAT .EQ. 0) GO TO 68
      WRITE(5,950) ISTAT
      PAUSE 12
      GO TO 60
C
C LOAD TIME OUT CONSTANT(TOC) FOR READING. FIRST HALFWORD
C IS TOC, SECOND HALFWORD MUST BE NONZERO TO LOAD VALUE.
C DEFAULT TOC IS 10 SECONDS.
C
68    PBLK(6) = 0
      PBLK(6) = 1 + TOC*Y'10000'
      M = 2 * N
      CALL SYSIO(PBLK, Y'C0', LU, DUMMY, 0, IADR)
      CALL ILBYTE (ISTAT, PBLK, 2)
      IF(ISTAT .NE. Y'88') WRITE (5, 945) ISTAT
C
C READ 2N BYTES FROM CHANNEL SPECIFIED BY IMDEADR INTO
C BUFFER AND CHECK STATUS OF THE ADC CARD.
C
      CALL SYSIO(PBLK, Y'58', LU, BUFFER, M, 0)
      CALL ILBYTE(ISTAT, PBLK, 2)
      IF (ISTAT .EQ. 0) GO TO 70
      FLAG = .TRUE.
      WRITE (5, 960) ISTAT
      GO TO 40
C
930   FORMAT(' STATUS DURING CLEAR OF BUS WAS:' ,2X,Z4)
```

```
      940  FORMAT(' IN SELECTING ADC ON NBS BUS.'/
          1 ' STATUS WAS:' ,2X,Z4)
       945 FORMAT(' STATUS JUST BEFORE READ OF A/D WAS:'/
          1          ,2X,Z4)
      950  FORMAT(' STATUS DURING LOADING MODE AND ADDRESS REG. WAS:' ,2X,Z4)
      960  FORMAT(' IN READ FROM ADC. STATUS WAS:' ,2X,Z4)
       70  CALL SYSIO(PBLK, Y'C0', LU, DUMMY, 0, Y'B0')
           IF (FLAG) GO TO 30
           RETURN
           END
```

3.4 Flowchart for ROUGHNES
MAIN PROGRAM



The operator is prompted to type the name of the file that will be created to hold the profile data.

The operator is prompted to type the I.D., the type of stylus instrument and its magnification settings, and the horizontal point spacing of the data. These parameters are then output to the console, the data file, and a print file.

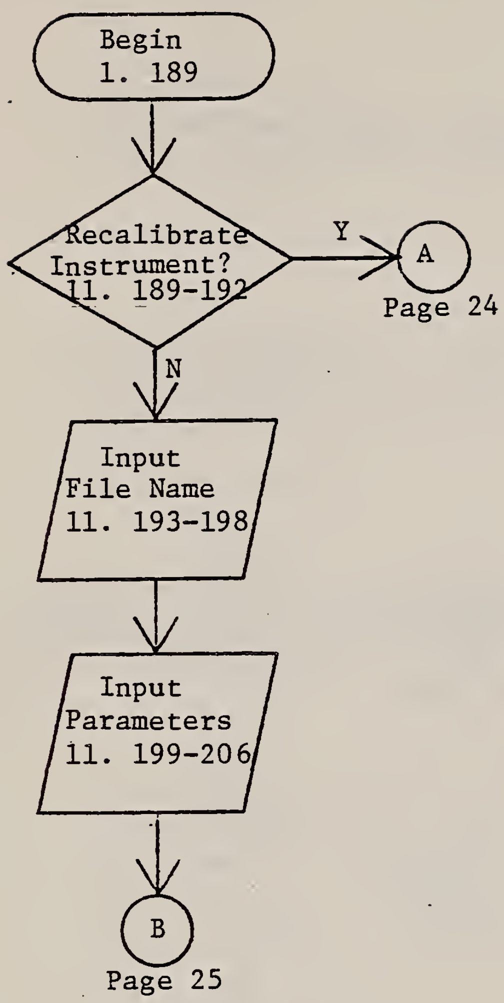
The NBS Bus is opened as a logical unit to the program. It functions as the interface between the computer and the stylus instrument.

The subroutine to measure the instrument calibration constant is called.

The subroutine to measure the roughness profiles and calculate R_a is called.

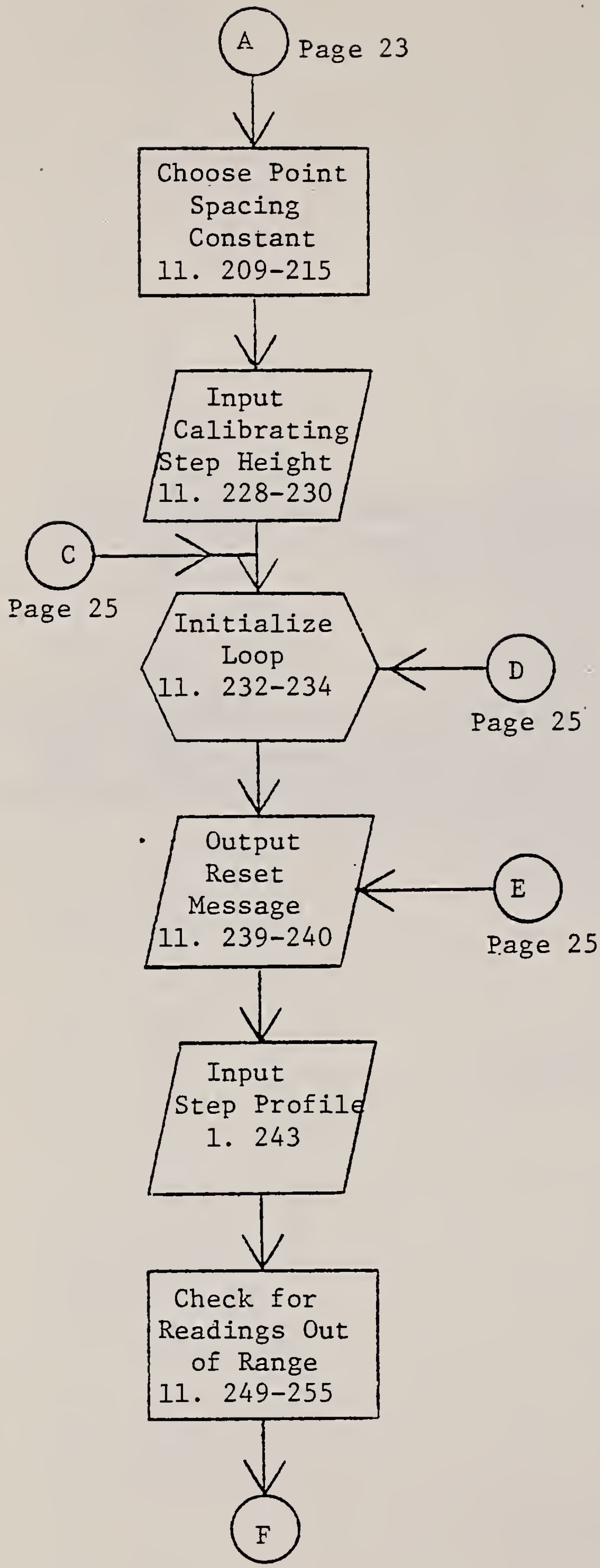
* "1.52" means "line 52"; "11. 52-61" means "lines 52-61".

KCAL1 SUBROUTINE



The operator is prompted to type the name of a previously created data file that holds the calibration data.

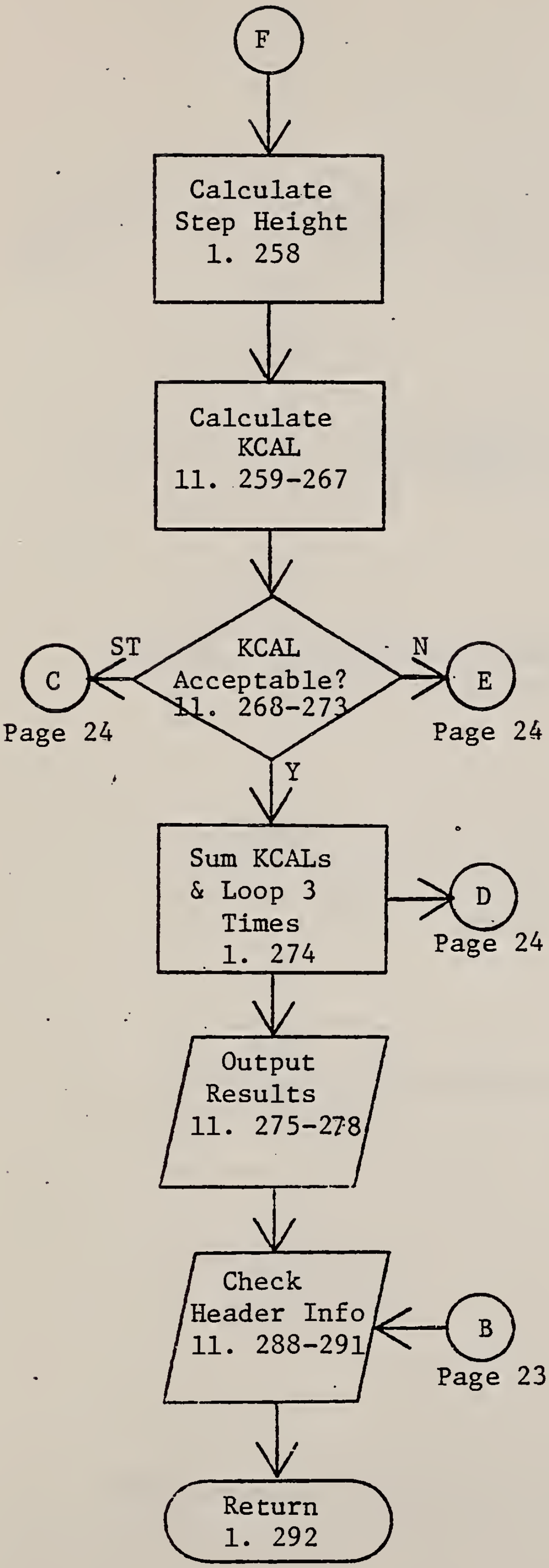
That file is read for the calibration constant KCAL and the calibrating step height. The values are then written into the current data file and the print file.



This constant depends on the chart speed and the data rate.

The operator is prompted to type the height of the calibrating step.

The calibrating step will be measured three times.



The subroutine STPHGT is called. It fits straight lines to the low and high sides of the step, extrapolates the lines to the middle of the step, and calculates the height difference in quantization levels at the middle of the step and at several other places along the profile.

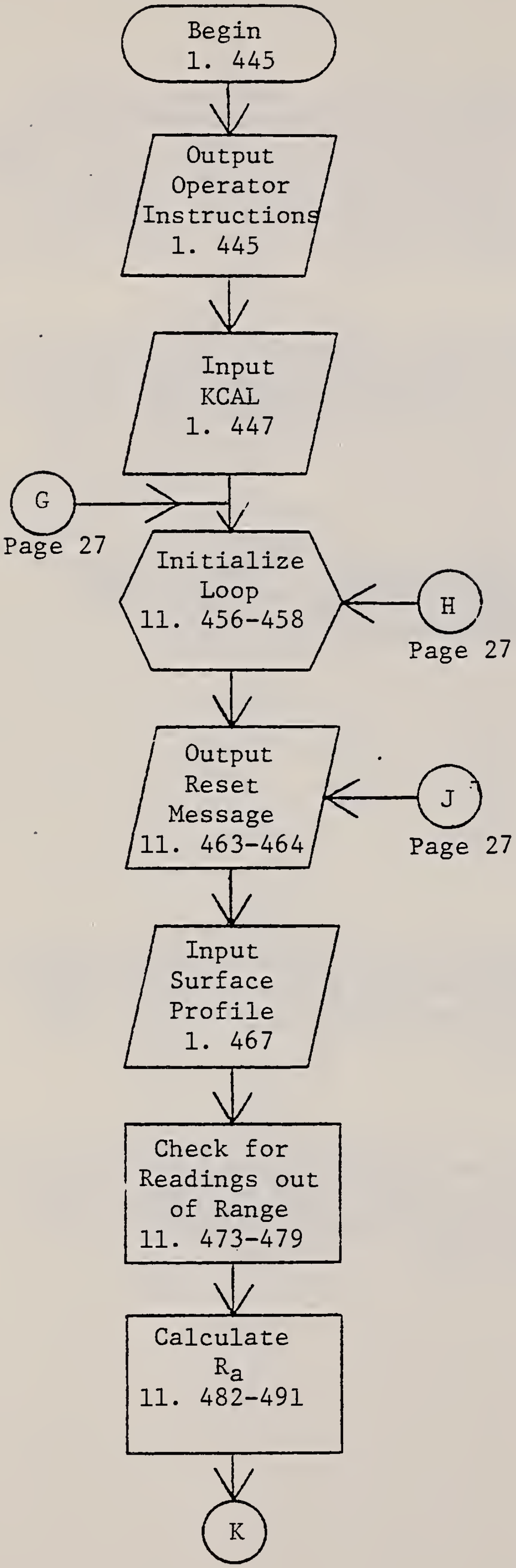
The calibration constant is calculated and displayed. The height differences discussed above are calculated and displayed in um.

If the operator types "ST", the program reinitializes the loop and begins the calibration procedure over again.

The average of three KCALs is calculated and written to the data file. KCAL and other step height parameters are written to the print file.

The first six lines of the data file are read back in and output to the console, so the operator can review the information, and to the print file.

RA1 SUBROUTINE

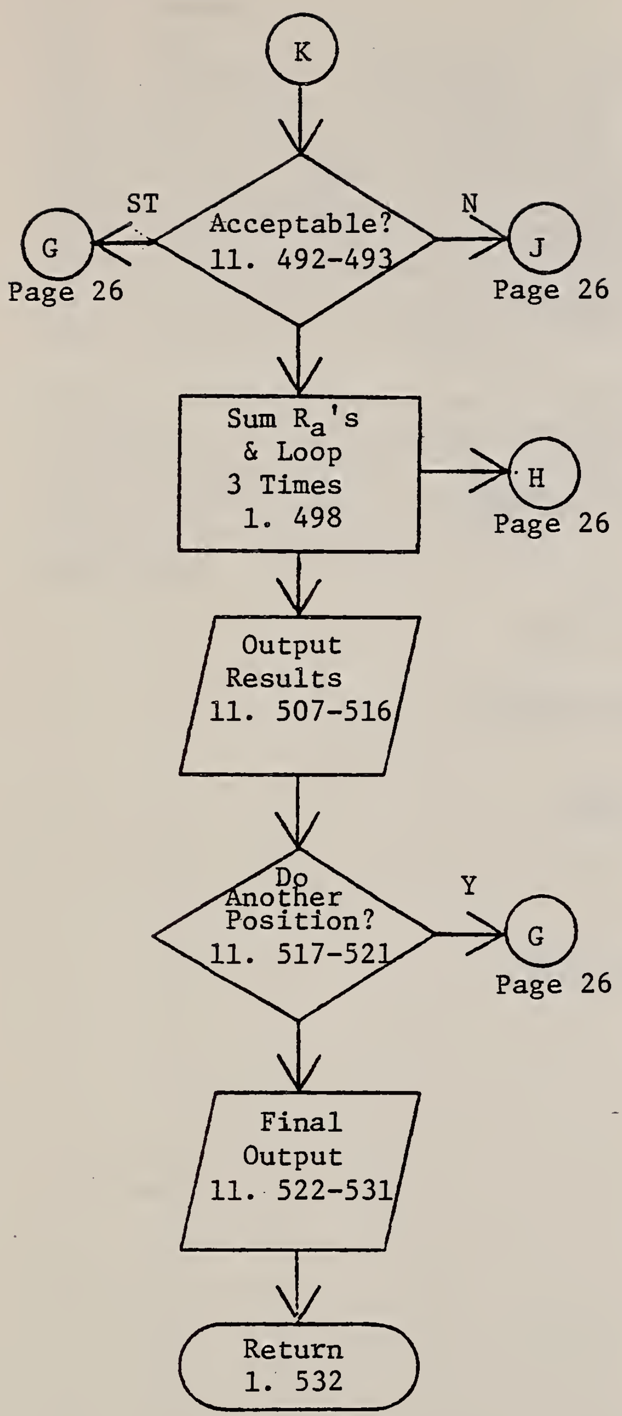


Read the calibration constant from the data file.

Each surface profile will be measured three times in this loop.

Subroutine ADCIO is called here.

The result is output to the console.



The average R_a is calculated. The operator may type in a label, which is then stored in the data file along with the third profile for each position. The measured R_a and other relevant information is output to the data file and the printer.

The final number of measurement positions is written on the data file and the print file is output to the printer.

3.5 Example of ROUGHNES Printout

```
NBS COMPUTERIZED SURFACE ROUGHNESS FACILITY
SD8:B27SEP82.DAT
TEST OF TALYSURF WITH 3 UM PROTOTYPE SRM
TALYSURF  2,000X VERT.  20X HORIZ.
THE POINT-TO-POINT SPACING IS  12.7300  UM.

          TRACE  1
          THE EXTREMA ARE 52 AND  62 MM.
              H1         H3         H5         H7
             12.7358    12.7319    12.7281    12.7242
          STEP HEIGHT =    12.7300 UM
          KCAL =  0.664203E-02 UM/QUANTIZATION LEVEL
          TRACE  2
          THE EXTREMA ARE 54 AND  60 MM.
              H1         H3         H5         H7
             12.7321    12.7307    12.7293    12.7279
          STEP HEIGHT =    12.7300 UM
          KCAL =  0.660992E-02 UM/QUANTIZATION LEVEL
          TRACE  3
          THE EXTREMA ARE 52 AND  60 MM.
              H1         H3         H5         H7
             12.7326    12.7309    12.7291    12.7274
          STEP HEIGHT =    12.7300 UM
          KCAL =  0.660349E-02 UM/QUANTIZATION LEVEL


THE FIRST SIX RECORDS IN TODAY"S DATA FILE ARE:
SD8:B27SEP82.DAT
TEST OF TALYSURF WITH 3 UM PROTOTYPE SRM
TALYSURF  2,000X VERT.  20X HORIZ.
  12.7300 UM/PT(SPACING)
  12.7300 UM = THE HEIGHT OF THE CALIBRATING STEP.
0.661848E-02 UM/QUANTIZATION LEVEL = KCAL



SURFACE ROUGHNESS RA MEASUREMENTS

POS.  1   AVERAGE RA =   2.9968 UM;
TRACE   1     RA =    3.0215  UM
TRACE   2     RA =    2.9616  UM
TRACE   3     RA =    3.0074  UM
```

4. STEPHGHT

4.1 Summary

The program STEPHGHT serves an important function in calibrations. It controls the data acquisition of stepped profiles and calculates the step heights. The main program is almost identical to the main program in ROUGHNES. It calls the subroutine STEPCAL, the first part of which calculates a calibration constant KCAL by calling ADCIO and STPHGT. The last part of STEPCAL then measures unknown steps. Each step position is measured three times, an average is calculated, and the third step profile is stored on the disk.

Nearly all of the procedures and options in STEPHGHT are similar to those of ROUGHNES. A key difference is that the surface profile consists of 1000 rather than 4000 data points. However, the profiles are stored in a 4000 point array with 3000 trailing zeros so that both kinds of profiles can be read by the same programs.

4.2 Operating System Commands

```
$***** STEPHGHT *****
LO .BG, SRF:STEPHGHT.TSK
T .BG
CLOSE ALL
AS 4,L4:
AS 5,L1:
AS 6,C:
XDE SRF:SPILL.DAT
AL SRF:SPILL.DAT,IN,80/5/5
AS 8,SRF:SPILL.DAT
$W* YOU SHOULD NOW BE RECEIVING INFORMATION ON THE SURFACE CONSOLE.
ST
*
* RETURN TO "FASTMENU".
*
FASTMENU
$EXIT
```

4.3 STEPHGHT FORTRAN Program

```
C ***** SRF:STEPHGHT.FTN CALLED BY "STEPHGHT" *****
C
C THIS PROGRAM BEGINS THE SERIES FOR STEP HEIGHT
C MEASUREMENTS. THE MAIN PROGRAM ALLOWS LABELING DATA CON-
C CERNING THE DATE, THE SPECIMEN, AND THE INSTRUMENT TO BE READ
C INTO A FILE WITH THE FORMAT "VOL:ADATE.DAT". THIS FILE IS
C CREATED BY THE OPERATOR AT THE BEGINNING OF THE MAIN
C PROGRAM AND WILL EVENTUALLY CONTAIN ALL OF THE PROFILE
C DATA GENERATED DURING THE RUN.
C       T. VORBURGER 4/13/78 (REVISED 9/82)
C
C
C
        INTEGER STAT1,STAT2,STAT,FDBUS(2)
        INTEGER*2 DATFIL(8),Q1,BUFFER(4000)
        COMMON /FUDGE/BUFFER
        DIMENSION SPEC(20),RINSTR(20)
        DATA DATFIL(7),DATFIL(8)/'.D','AT'/
        DATA FDBUS/'BUS:   '/
C
C
1       FORMAT(' ***** NBS COMPUTERIZED SURFACE ROUGHNESS FACILITY',
       1' IS READY TO GO. *****'///
       1' TYPE THE NAME OF THE DATA FILE FOR TODAY IN THE' /
       2' FOLLOWING FORMAT: "VOL:ADATE", WHERE VOL IS THE' /
       3' THREE CHARACTER VOLUME NAME AND ADATE IS THE EIGHT' /
       4' CHARACTER FILE NAME.' )
2       FORMAT(1H ,20A4)
3       FORMAT(1H1/' NBS COMPUTERIZED SURFACE ROUGHNESS FACILITY,',
       1' STEP HEIGHT MEASUREMENTS')
4       FORMAT(' TYPE THE SPECIMEN ID INFORMATION.' )
5       FORMAT(20A4)
7       FORMAT(1H ,9A2)
8       FORMAT(' TYPE "TALYSTEP" OR "TALYSURF" AND THE VERTICAL' /
       1' AND HORIZONTAL MAGNIFICATIONS.' )
9       FORMAT(' TYPE THE POINT-TO-POINT SPACING IN UM/PT.' /
       1' INCLUDE THE DECIMAL POINT EXPLICITLY.' )
10      FORMAT(F10.4)
11      FORMAT(' THE FIRST FOUR RECORDS IN ' ,8A2/
       1' ARE AS FOLLOWS:' )
12      FORMAT(8A2)
13      FORMAT(' THE POINT-TO-POINT SPACING IS ' ,F8.4,
       1'  UM.' )
17      FORMAT(' ERROR STATUS = ' ,I2//' COME ON TURKEY.',
       1' DO IT OVER AND GET IT RIGHT THIS TIME!' )
19      FORMAT(' NBS BUS ASSIGN ERROR. STATUS WAS:' ,2X,I4)
C
C
C FIRST, THE OPERATOR IS ASKED TO TYPE THE FILENAME, THAT WILL
C BE CREATED TO HOLD THE DATA, AND THE SAMPLE IDENTIFICATION.
C
C
92      WRITE(5,1)
        READ(5,12) (DATFIL(J),J=1,6)
        CALL CFILW(DATFIL,2,80,5,5,0,0,STAT1)
        IF (STAT1.LT.1) GO TO 90
        WRITE(5,17) STAT1
        GO TO 92
```

```
90    CALL OPENW(1,DATFIL,4,0,0,STAT2)
C
C
C NOW, THE OPERATOR IS ASKED TO TYPE THE HEADER INFOR-
C MATION CONCERNING THE SPECIMEN, THE INSTRUMENT, AND
C THE SPACING OF THE DATA POINTS.
C
C
      WRITE(5,4)
      READ(5,5) SPEC
      WRITE(5,8)
      READ(5,5) RINSTR
      WRITE(5,9)
      READ(5,10) PTTOPT
      WRITE(5,11) DATFIL
      WRITE(5,7) DATFIL
      WRITE(5,2) SPEC
      WRITE(5,2) RINSTR
      WRITE(5,10) PTTOPT
      WRITE(1,12) DATFIL
      WRITE(1,5) SPEC
      WRITE(1,5) RINSTR
      WRITE(1,10) PTTOPT
      WRITE(8,3)
      WRITE(8,7) DATFIL
      WRITE(8,2) SPEC
      WRITE(8,2) RINSTR
      WRITE(8,13) PTTOPT
C
C
C AT THIS POINT WE OPEN THE NBS BUS AS LU-3.
C
C
50    CALL OPENW(3,FDBUS,4,0,0,STAT)
      IF (STAT .EQ. 0) GO TO 105
      WRITE (5,19) STAT
      PAUSE 1
      GO TO 50
C
C
C FINALLY, THIS PROGRAM CALLS THE STEPCAL SUBROUTINE.
C
C
105   CALL STEPCAL(DATFIL)
      CLOSE(1)
      CLOSE(3)
      CLOSE(8)
      STOP
      END
C
C
C
C
C
      SUBROUTINE STEPCAL(DATFIL)
C THIS ROUTINE CALCULATES THE CALIBRATION CONSTANT AND STEP
C HEIGHTS FROM THE PROFILE DATA.  ALL RESULTS
C ARE RECORDED IN THE FILE "VOL:ADATE.DAT".
```

```
C                                                       TVTV 4/17/78
C
C
       INTEGER*2 DATFIL(8),OLDFIL(8),Q1,Q2,Q3,Q4,Q5
       DIMENSION SPEC(20),RINSTR(20),TALY(2),COMENT(10)
       DIMENSION NKCAL(3),IE1(3),IE2(3),H1(3),H3(3),H5(3)
       DIMENSION H7(3),H4(3),XSTEP(3),REGURG(20)
       REAL KCAL(3)
       INTEGER*2 BUFFER(4000)
       INTEGER NTRACE(3)
       INTEGER STAT,STAT2,STAT4,STATUS,FDBUS(2)
       COMMON /FUDGE/BUFFER
       DATA IBLANK/'    '/
       DATA OLDFIL(7),OLDFIL(8)/'.D','AT'/
       DATA FDBUS/'BUS:   '/
C
 1     FORMAT(' WE NOW BEGIN THE STEP MEASUREMENT OR RECAL',
      1'IBRATION PROCEDURE.')
 2     FORMAT(' WE WILL NOW START THE RECALIBRATION PROCEDURE.'/
      1' WHAT IS THE HEIGHT OF THE CALIBRATING STEP IN UM?'/
      2' INCLUDE THE DECIMAL POINT EXPLICITLY.')
 3     FORMAT(40A2)
 4     FORMAT(' DO YOU WISH TO RECALIBRATE THE INSTRUMENT?'/
      1' "Y" OR "N"?')
 5     FORMAT(20A4)
 6     FORMAT(1H1/' ADJUST THE FREQUENCY GENERATOR TIMING '/
      1' FOR STEP MEASUREMENT ON THE ***** ',2A4,' *****,'/
      2' AND SET THE FILTER FOR "STEPS" MODE.'///)
 7     FORMAT(F10.4)
 8     FORMAT(1H ,20A4)
 9     FORMAT(4X,'H1',8X,'H3',8X,'H5',8X,'H7'/4F10.4)
 10    FORMAT(/' POSITION',I3,'     TRACE',I3,
      1'   STEP HEIGHT =',F10.4,' UM')
 11    FORMAT(' KCAL = ',E13.6,' UM/QUANTIZATION LEVEL')
 12    FORMAT(' LOOK OK?'/' TYPE "YES" IF IT LOOKS OK,'/
      1' TYPE "NO" IF YOU WANT TO REDO THIS TRACE,'/
      2' AND TYPE "ST" IF YOU WANT TO START THE KCAL SEQUENCE ',
      3'OR THE POSITION OVER.')
 13    FORMAT(3(/10X,'TRACE',I3/10X,'THE EXTREMA ARE',I3,' AND ',
      1I3,' MM.'/10X,4X,'H1',8X,'H3',8X,'H5',8X,'H7'/10X,4F10.4/10X,
      2'STEP HEIGHT = ',F10.4,' UM'/10X,'KCAL = ',E13.6,
      3' UM/QUANTIZATION LEVEL'))
 14    FORMAT(E13.6)
 15    FORMAT(9A2)
 16    FORMAT(//' THE FIRST SIX RECORDS IN TODAY"S FILE ARE:'/
      11X,8A2/1X,20A4/1X,20A4/F10.4,' UM/PT(SPACING)'/F10.4,' UM =',
      2' THE HEIGHT OF THE CALIBRATING STEP.'/E13.6,
      3' UM/QUANTIZATION LEVEL = KCAL')
 17    FORMAT(20A4/20A4/F10.4/F10.4/E13.6)
 18    FORMAT(//'  STEP MEASUREMENT PROCEDURE:'/' WE WILL MEASURE THE'
      1' STEP THREE TIMES AT EACH POSITION AND TAKE THE AVERAGE.'/
      2'*****ADJUST THE TIMING.')
 19    FORMAT(' BAD DATA. YOU CHOOSE TO START THE POSITION ',
      1' OR THE KCAL SEQUENCE OVER.')
 21    FORMAT(/' POS.',I3,'  AVE. STEP HEIGHT =',F8.4,
      1' UM; ',10A4,3(/' TRACE ',I3,'   STEP HEIGHT =',F10.4,' UM'))
 22    FORMAT(' DO ANOTHER POSITION?')
 23    FORMAT(I2,F10.4,10A4)
```

```
24    FORMAT(' END OF STEP HEIGHT ROUTINE.')
25    FORMAT(' ***** RESET STYLUS INSTRUMENT.'/
     1' ***** THEN HIT RETURN KEY, WHEN YOU ARE READY TO TAKE DATA.')
26    FORMAT(16I5)
28    FORMAT(/' KCAL ROUTINE  TRACE',I3,
     1'  STEP HEIGHT = ',F10.3,' UM')
29    FORMAT(' WHAT IS THE NAME OF THE FILE THAT YOU ',
     1'WANT THE KCAL FROM?'/' USE THE TWELVE CHARACTER',
     2' FORMAT "VOL:ADATE",'/' WHERE "VOL" IS THE THREE',
     3' CHARACTER VOLUME NAME'/' AND "ADATE" IS THE',
     4' EIGHT CHARACTER FILE NAME.')
30    FORMAT(' ERROR CODE',I3/' SOMETHING"S WRONG.',
     1' I CAN"T OPEN THIS FILE.'/' TRY AGAIN.')
31    FORMAT(' DATA READING COMPLETE.'/1X,I4,2X,' POINTS',
     1' OVERFLOW.'/1X,I4,2X,' POINTS UNDERFLOW.')
32    FORMAT(/////' THE CALIBRATION CONSTANT KCAL HAS ',
     1'BEEN OBTAINED FROM THE FILE ',9A2)
33    FORMAT(/' WE NOW START THE RECALIBRATION PROCEDURE.')
34    FORMAT(' CALIBRATING STEP = ',F10.4,' UM'/////)
35    FORMAT(' DO YOU HAVE ANY COMMENTS FOR THIS POSITION?'/
     1' YOU ARE ALLOWED 40 CHARACTERS.'/ ' IF NOT, JUST HIT "RETURN".')
36    FORMAT(10A4)
37    FORMAT(1H1)
C
C
      WRITE(5,1)
      READ(1,5,REC=3)TALY
      WRITE(5,6) TALY
      IF(TALY(2).EQ.'SURF') GO TO 104
      HORSPC = 10.
      GO TO 105
104   HORSPC = 6.6667
C
C
C IN THE NEXT SEQUENCE, THE OPERATOR IS ASKED TO DECIDE
C WHETHER TO RECALIBRATE THE INSTRUMENT OR TO USE A
C CALIBRATION CONSTANT FROM ANOTHER FILE.
C
C
105   WRITE(5,4)
      READ(5,3) Q1
      IF(Q1.EQ.'N') GO TO 120
      GO TO 126
120   WRITE(5,29)
      READ (5,3) (OLDFIL(J),J=1,6)
      CALL OPENW (2,OLDFIL,0,0,0,STAT2)
      IF (STAT2 .LT. 1) GO TO 121
      WRITE (5,30) STAT2
      GO TO 120
121   READ (2,7,REC=5) CALSTP
      READ (2,14) AVKCAL
      WRITE (8,32) (OLDFIL(J),J=1,6)
      WRITE(8,34) CALSTP
      CALL CLOSE (2,STAT2)
      WRITE (1,7,REC=5) CALSTP
      WRITE (1,14) AVKCAL
      GO TO 127
C
```

```
C
C KCAL ROUTINE: THE DATA ARE READ INTO THE COMPUTER VIA
C THE ADCIO ROUTINE. THE RESULTING UNCALIBRATED STEP
C PROFILE IS FITTED BY TWO STRAIGHT LINES  IN THE STEP
C FITTING SUBROUTINE,STPHGT. THE RESULT IS PASSED BACK TO
C THE MAIN PROGRAM AS THE VARIABLE Y4, WHICH IS DIVIDED
C BY THE HEIGHT OF THE CALIBRATING STEP TO YIELD A VALUE
C FOR KCAL IN UM/QUANTIZATION LEVEL.
C
C
126   WRITE(5,2)
      WRITE (8,33)
      READ(5,7) CALSTP
      WRITE(1,7,REC=5) CALSTP
C
110     SUM1 = 0.
        DO 102 K=1,3
        NKCAL(K) = K
C
C
C AT THIS POINT WE REMIND THE OPERATOR TO RESET THE INSTRUMENT.
C
103     WRITE (5,25)
        READ (5,5) JUNK
C
C
        CALL ADCIO(1000,Y'2307',4,3,BUFFER,40)
C
C
C INFORM OPERATOR OF COMPLETION OF DATA READING AND AMOUNT
C OF UNDERFLOW AND OVERFLOW.
C
        L1 = 0
        L2 = 0
        DO 75 I = 1, 1000
        IF (BUFFER(I) .EQ. 2047) L1 = L1 + 1
        IF (BUFFER(I) .EQ. -2048) L2 = L2 + 1
75      CONTINUE
        WRITE (5, 31) L1, L2
C
C
        CALL STPHGT(Y1,Y3,Y4,Y5,Y7,IE1(K),IE2(K),HORSPC)
        KCAL(K) = CALSTP/Y4
        H1(K) = KCAL(K)*Y1
        H3(K) = KCAL(K)*Y3
        H4(K) = KCAL(K)*Y4
        H5(K) = KCAL(K)*Y5
        H7(K) = KCAL(K)*Y7
        WRITE(5,9) H1(K),H3(K),H5(K),H7(K)
        WRITE (5,28) K,H4(K)
        WRITE(5,11) KCAL(K)
        WRITE(5,12)
        READ(5,3) Q2
        IF (Q2 .EQ. 'NO') GO TO 103
        IF (Q2 .NE. 'ST') GO TO 102
        WRITE (5,19)
        GO TO 110
        IF( Q2.EQ.'NO') GO TO 103
```

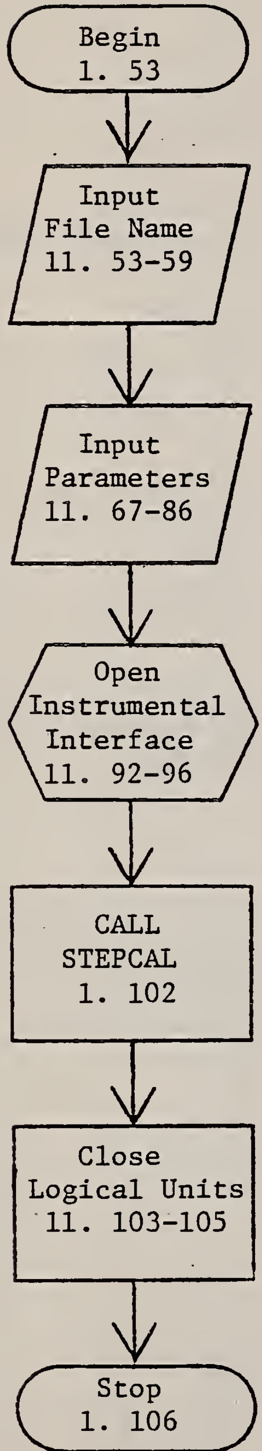
```
102     SUM1 = SUM1+KCAL(K)
C
C
C THE AVERAGE VALUE FOR KCAL(K) IS STORED IN AVKCAL AND
C WRITTEN INTO THE FILE "VOL:ADATE.DAT". THEN ALL OF
C THE LABELING INFORMATION CONTAINED IN THE FILE IS
C READ BACK AND PRINTED OUT FOR THE OPERATOR TO CHECK.
C THIS INCLUDES THE FILE NAME, THE SPECIMEN AND INSTRUMENT
C INFORMATION, THE POINT-TO-POINT SPACING, THE CALIBRATING
C STEP HEIGHT, AND KCAL(K).
C
C
      AVKCAL = SUM1/3.
      WRITE(1,14) AVKCAL
      WRITE(8,13) (NKCAL(K),IE1(K),IE2(K),H1(K),H3(K),
     1H5(K),H7(K),H4(K),KCAL(K),K=1,3)
127   READ(1,15,REC=1)DATFIL
      READ(1,17)SPEC,RINSTR,PTTOPT,CALSTP,AVKCAL
      WRITE(5,16) DATFIL,SPEC,RINSTR,PTTOPT,CALSTP,AVKCAL
      WRITE(8,16) DATFIL,SPEC,RINSTR,PTTOPT,CALSTP,AVKCAL
C
C
C STEP MEASUREMENT SEQUENCE: KCAL IS READ FROM THE FILE
C "VOL:ADATE.DAT". THEN, THE VARIABLE NPOS, WHICH GIVES
C THE NUMBER OF POSITIONS MEASURED IS INITIALIZED AND RECORDED
C IN "VOL:ADATE.DAT".
C
C
      WRITE(5,18)
      READ(1,14,REC=6) AVKCAL
      NPOS = 1
      WRITE(1,3) NPOS
C
C
C NOW THE STEP PROFILE DATA ARE READ INTO THE COMPUTER
C VIA THE ADCIO ROUTINE. THE RESULTING UNCALIBRATED
C STEP PROFILE IS FITTED BY TWO STRAIGHT LINES IN THE
C STEP FITTING SUBROUTINE,STPHGT. THE RESULT IS PASSED
C BACKED TO THIS PROGRAM AS THE VARIABLE Y4, WHICH IS
C MULTIPLIED BY AVKCAL TO YIELD A VALUE FOR THE STEP
C HEIGHT, XSTEP, IN UM. THIS IS DONE THREE TIMES FOR
C EACH POSITION. THE OPERATOR HAS THE OPTION OF
C ACCEPTING OR REJECTING EACH XSTEP OR OF BEGINNING
C THE POSITION OVER.
C
C
204     SUM3 = 0.
        DO 201 K=1,3
        NTRACE(K) = K
C
C
C AT THIS POINT WE REMIND THE OPERATOR TO RESET THE INSTRUMENT.
C
202     WRITE (5,25)
        READ (5,5) JUNK
C
C
        CALL ADCIO(1000,Y'2307',4,3,BUFFER,40)
```

```
C
C
C INFORM OPERATOR OF COMPLETION OF DATA READING AND AMOUNT
C OF UNDERFLOW AND OVERFLOW.
C
        L1 = 0
        L2 = 0
        DO 76 I = 1, 1000
        IF (BUFFER(I) .EQ. 2047) L1 = L1 + 1
        IF (BUFFER(I) .EQ. -2048) L2 = L2 + 1
76      CONTINUE
        WRITE (5, 31) L1, L2
C
C
        CALL STPHGT(Y1,Y3,Y4,Y5,Y7,IE1(K),IE2(K),HORSPC)
        XSTEP(K) = AVKCAL*Y4
        H1(K) = AVKCAL*Y1
        H3(K) = AVKCAL*Y3
        H5(K) = AVKCAL*Y5
        H7(K) = AVKCAL*Y7
        WRITE(5,9) H1(K),H3(K),H5(K),H7(K)
        WRITE(5,10) NPOS,NTRACE(K), XSTEP(K)
        WRITE(5,12)
        READ(5,3) Q3
        IF (Q3.EQ.'NO') GO TO 202
        IF (Q3.NE.'ST') GO TO 201
        WRITE(5,19)
        GO TO 204
201     SUM3 = SUM3+XSTEP(K)
C
C
C THE AVERAGE STEP HEIGHT IS NOW CALCULATED AND THE RESULTS
C ARE WRITTEN ON THE SURFACE CONSOLE, THE PRINTER, AND THE
C FILE "VOL:ADATE.DAT". THE PROFILE DATA FROM TRACE 3 FOR
C EACH POSITION ARE ALSO WRITTEN INTO THE DATA FILE. THE
C OPERATOR HAS THE OPTION OF GOING TO A NEW POSITION OR
C OF ENDING THE PROGRAM.
C
C
      AVSTEP = SUM3/3.
      WRITE(5,35)
      READ(5,36) COMENT
      WRITE(1,23) NPOS,AVSTEP,COMENT
      WRITE(1,26) BUFFER
      IF (NPOS.EQ.6 .OR. NPOS.EQ.16 .OR. NPOS.EQ.26
     1.OR. NPOS.EQ.36 .OR. NPOS.EQ.46 .OR. NPOS.EQ.56
     2.OR. NPOS.EQ.66 .OR. NPOS.EQ.76 .OR. NPOS.EQ.86) WRITE(8,37)
      WRITE(8,21) NPOS,AVSTEP,COMENT,(NTRACE(K),XSTEP(K),K=1,3)
      WRITE(5,22)
      READ(5,3) Q4
      IF (Q4.EQ.'NO') GO TO 206
      NPOS = NPOS+1
      GO TO 204
206   WRITE(1,23,REC=7) NPOS
      REWIND 8
207   READ(8,5,END=208) REGURG
      IF (REGURG(1).NE.'1   ') GO TO 209
      WRITE(4,37)
```

```
      GO TO 207
209   WRITE(4,S) REGURG
      GO TO 207
208   WRITE(5,24)
      RETURN
      END
```

The remaining subroutines, STPHGT and ADCIO, are shown in the ROUGHNES program.

4.4 Flowchart for STEPHGHT
MAIN PROGRAM



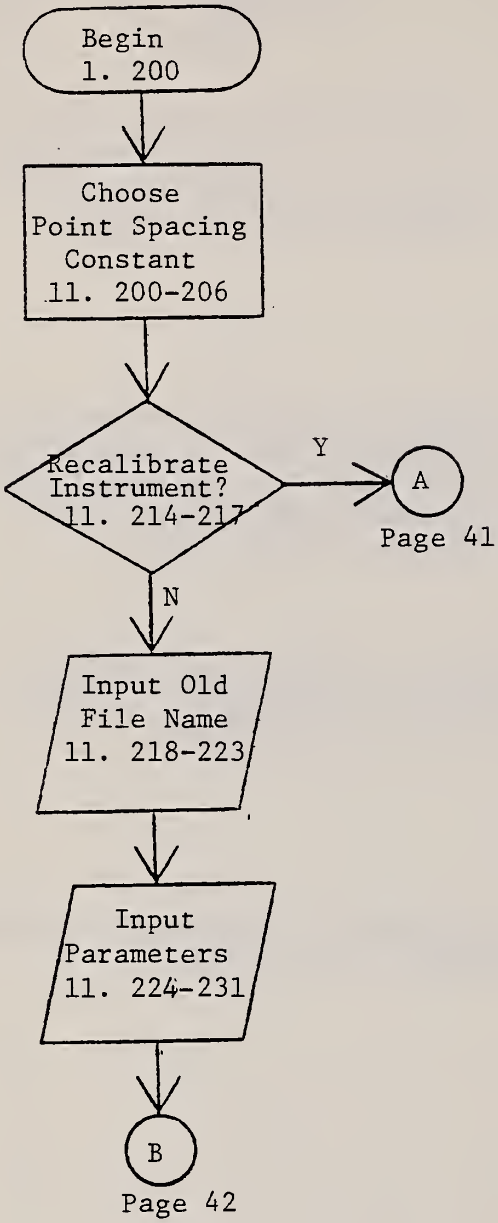
The operator is prompted to type the name of the file that will be created to hold the profile data.

The operator is prompted to type the specimen I.D., the type of stylus instrument and its magnification settings, and the horizontal point spacing of the data. These parameters are then output to the console, the data file, and a print file.

The NBS Bus is opened as a logical unit to the program. It functions as the interface between the computer and the stylus instrument.

The subroutine to determine the calibration constant and measure step heights is now called.

STEPCAL SUBROUTINE

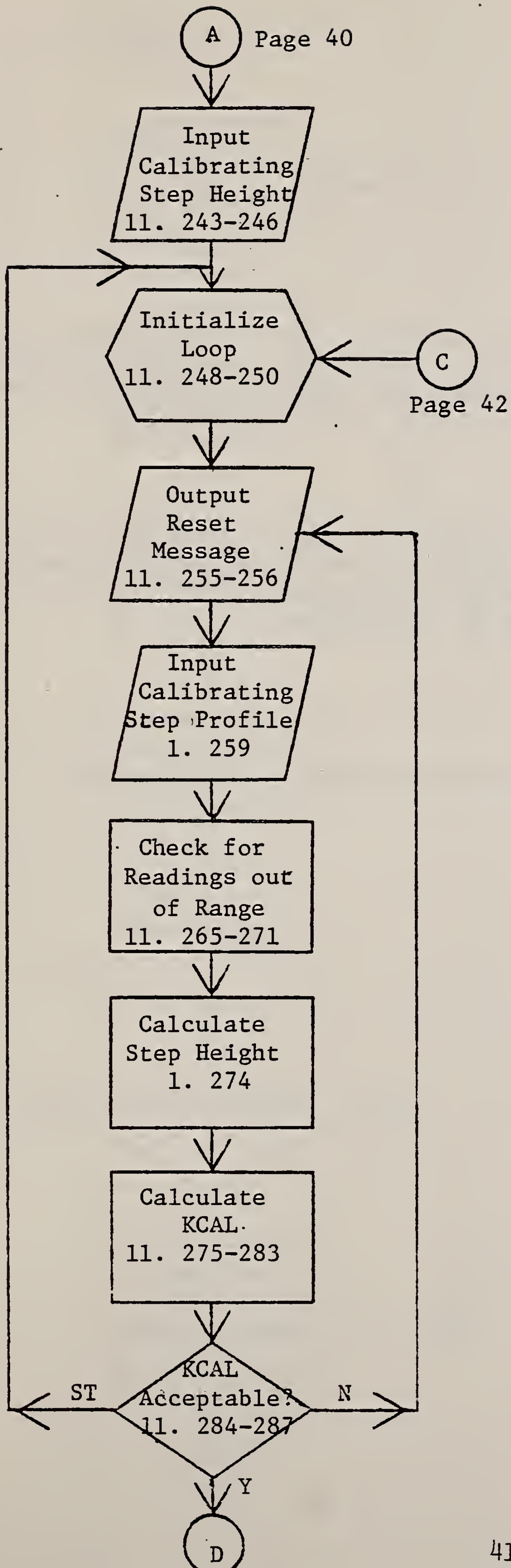


This constant depends on the chart speed and the data input rate.

The operator is prompted to make this choice.

The operator is then asked to type the name of a previously created data file that holds the calibration data.

That file is then read for the calibration constant KCAL and calibrating step height. The values are also written into the current data file and the print file.



The operator is prompted to type in the height of the calibrating step.

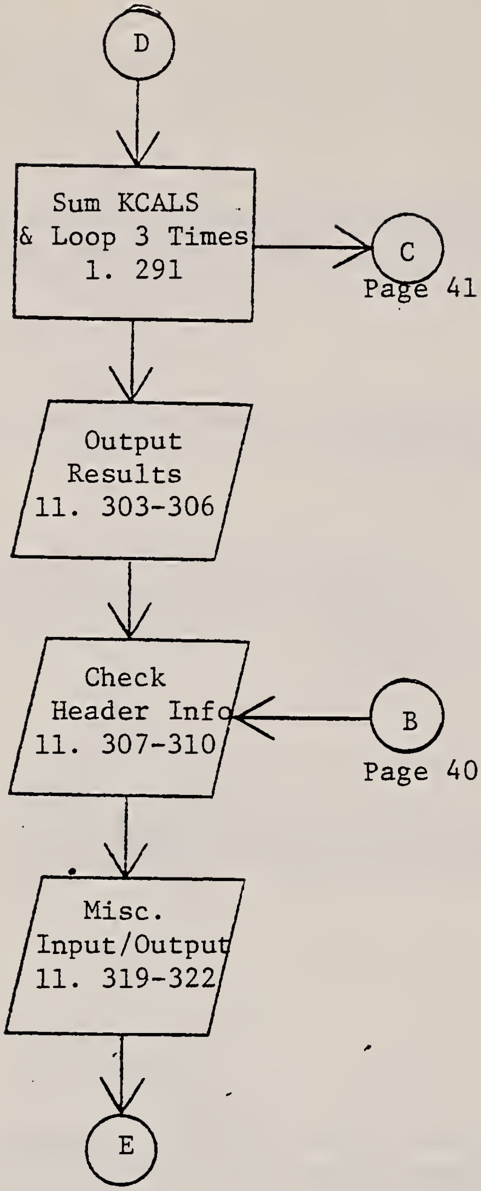
The calibrating step will be measured three times.

Subroutine ADCIO is called here.

The subroutine STPHGT is called. It fits straight lines to the low and high sides of the step, extrapolates the lines to the middle of the step, and calculates the height difference in quantization levels at the middle of the step and at several other places along the profile.

The calibration constant KCAL is calculated and displayed. The height differences discussed above are calculated and displayed in um.

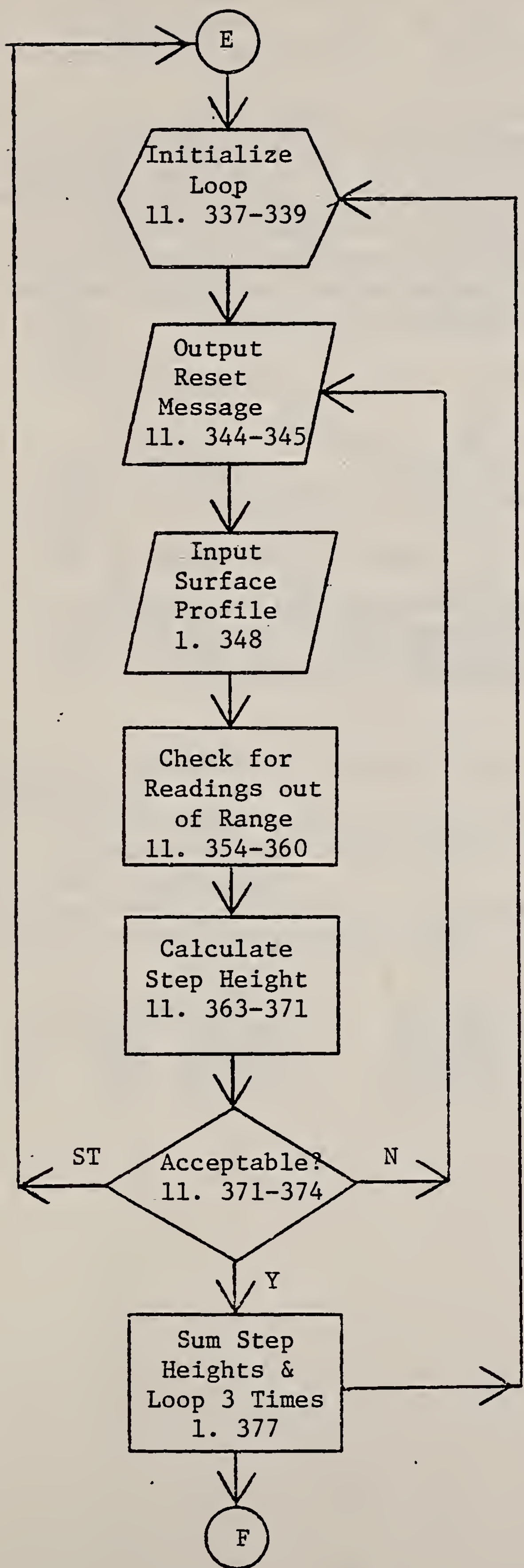
If the operator types "ST", the program reinitializes the loop and begins the calibration procedure over again.



The average of 3 KCALs is calculated and written to the data file. The calibration constant and other step height parameters are written to the print file.

The first six lines of the data file are read back in and output to the console, so the operator can review the information, and to the printer.

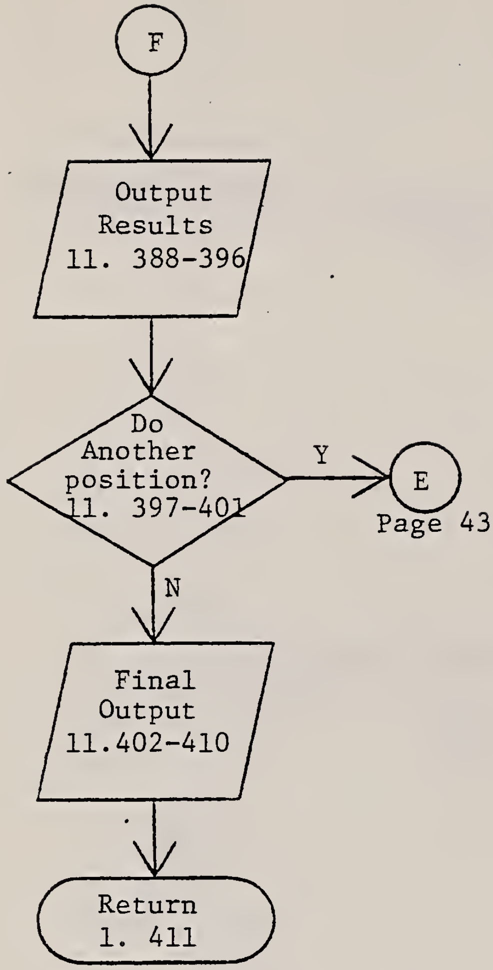
File manipulation and instructions.



Each Step profile will be measured three times.

Subroutine ADCIO is called.

The subroutine STPHGT is called again. This time, the results are used to calculate the height of the unknown step under test. The results are displayed on the console.



The average step height is calculated. At this point the operator may type in a label, which is then stored in the data file along with the third profile for each position. The measured step height (and other relevant information) is output to the data file and to the print file.

The final number of measured profiles is written on the data file, and the print file is output to the printer.

4.5 Example of STEPHGHT Printout

```
NBS COMPUTERIZED SURFACE ROUGHNESS FACILITY, STEP HEIGHT MEASUREMENTS
SD8:C27SEP82.DAT
TALYSTEP SYSTEM CHECK USING THE 2.993UM CALIBRATION STEP
TALYSTEP  10,000X VERT.   50X HORIZ.
THE POINT-TO-POINT SPACING IS   2.0000  UM.

WE NOW START THE RECALIBRATION PROCEDURE.

         TRACE  1
         THE EXTREMA ARE 32 AND  54 MM.
             H1         H3         H5         H7
             2.9875     2.9912     2.9948     2.9985
         STEP HEIGHT =      2.9930 UM
         KCAL =  0.127003E-02 UM/QUANTIZATION LEVEL
         TRACE  2
         THE EXTREMA ARE 37 AND  60 MM.
             H1         H3         H5         H7
             2.9864     2.9908     2.9952     2.9996
         STEP HEIGHT =      2.9930 UM
         KCAL =  0.126280E-02 UM/QUANTIZATION LEVEL
         TRACE  3
         THE EXTREMA ARE 37 AND  61 MM.
             H1         H3         H5         H7
             2.9846     2.9902     2.9958     3.0014
         STEP HEIGHT =      2.9930 UM
         KCAL =  0.126439E-02 UM/QUANTIZATION LEVEL


THE FIRST SIX RECORDS IN TODAY"S FILE ARE:
SD8:C27SEP82.DAT
TALYSTEP SYSTEM CHECK USING THE 2.993UM CALIBRATION STEP
TALYSTEP  10,000X VERT.   50X HORIZ.
   2.0000 UM/PT(SPACING)
   2.9930 UM = THE HEIGHT OF THE CALIBRATING STEP.
0.126574E-02 UM/QUANTIZATION LEVEL = KCAL

POS.   1  AVE. STEP HEIGHT =  3.0002 UM; REMEASUREMENT OF 2.993 UM STEP
TRACE    1   STEP HEIGHT =    3.0000 UM
TRACE    2   STEP HEIGHT =    3.0085 UM
TRACE    3   STEP HEIGHT =    2.9922 UM
```

5. WHATSON

5.1 Summary

The operator can find out what kinds of data are stored on a disk by typing WHATSON followed by the name of the disk. The disk must first be in place in one of the drives and marked on by the operating system. The program reads the disk directory into a file named SRF:WHATSON.DAT and reads each line in this file for the names of the data files. Then it reads and prints the first ten logical records of each data file so that the operator can scan the information and learn the important parameters of each data file.

5.2 Operating System Commands

```
***** WHATSON *****
*
XDE SRF:WHATSON.DAT
AL SRF:WHATSON.DAT,IN,80
D T,SRF:WHATSON.DAT
D F,@1,SRF:WHATSON.DAT
LO .BG,SRF:WHATSON.TSK
T .BG
AS 1,SRF:WHATSON.DAT
AS 3,L4:
AS 6,C:
ST
* RETURN TO "FASTMENU"
*
FASTMENU
$EXIT
```

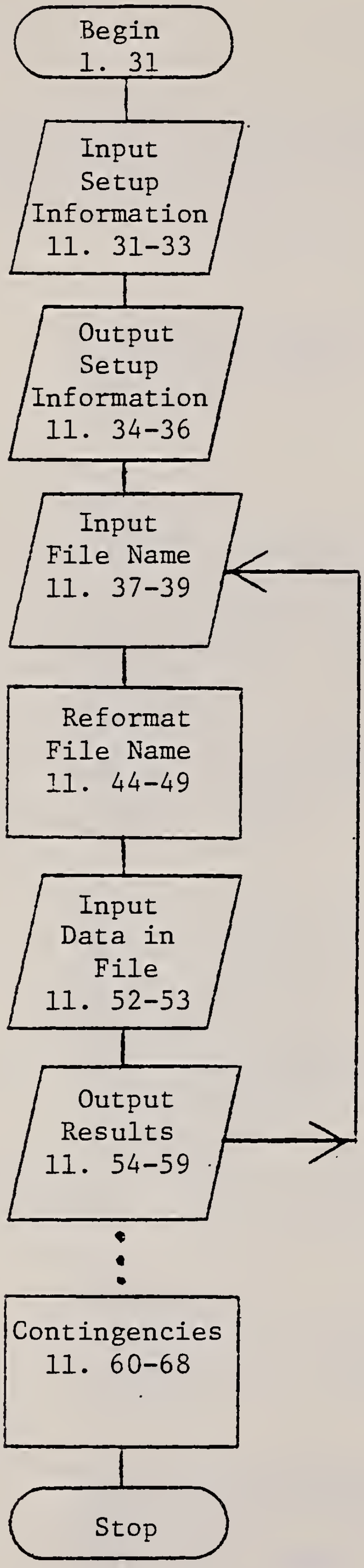
Notes:
"D T" writes the time and date in the temporary file SRF:WHATSON.DAT.
"D F" writes the file directory from the chosen disk in SRF:WHATSON.DAT.

5.3 WHATSON FORTRAN Program

```
C ***** SRF:WHATSON.FTN CALLED BY "WHATSON" *****
C THIS PROGRAM IS USED TO FIND OUT WHAT KINDS OF
C DATA ARE STORED ON A DISK . THE DISK IS SPECIFIED
C BY THE OPERATOR IN THE "WHATSON" COMMAND THAT
C STARTS THIS PROGRAM.
C           T. VORBURGER (8/82)
C
C
      INTEGER LABELS(20),STATUS,STAT2,WORDS(200),TIME(20)
      CHARACTER*80 HEADER
      CHARACTER*16 FILNAM
      HEADER(5:5) = ':'
C
C
1     FORMAT(A75)
2     FORMAT(' ')
3     FORMAT(/)
4     FORMAT(/' AN ERROR WAS ENCOUNTERED WHILE READING'/
     1' A LINE IN THE DIRECTORY.  I"LL TRY ANOTHER.')
5     FORMAT(' END OF FILE DIRECTORY'/' THAT"S ALL.')
6     FORMAT(8X,A4)
7     FORMAT(' HERE"S WHAT"S ON  DISK ',A5,' TODAY.')
8     FORMAT(1H1)
9     FORMAT(20A4)
10    FORMAT(/' I CAN"T OPEN THIS FILE:',A17/' I"LL TRY ANOTHER.')
11    FORMAT(/' I CAN"T READ THIS FILE:',A17/' I"LL TRY ANOTHER.')
13    FORMAT(1H ,20A4)
14    FORMAT(/1X,A75)
C
C
      READ(1,9) TIME
      READ(1,6) HEADER(1:4)
      READ(1,9) LABELS
      WRITE(3,8)
      WRITE(3,9) TIME
      WRITE(3,7) HEADER(1:5)
50    READ(1,1,END=102,ERR=101) HEADER(6:80)
      IF (HEADER(15:17) .NE. 'DAT') GO TO 50
      HEADER(14:14) = '.'
C
C THIS LOOP STRIPS OUT THE BLANKS FROM THE FILE
C NAME SO IT CAN BE ASSIGNED.
C
      J1=0
      DO 201 J=1,17
      IF (HEADER(J:J) .EQ. ' ') GO TO 201
      J1 = J1+1
      FILNAM(J1:J1) = HEADER(J:J)
201   CONTINUE
C
C
      OPEN(2,FILE=FILNAM(1:J1),STATUS='OLD',ERR=104)
      READ(2,9,ERR=105,END=105)  WORDS
      WRITE(3,3)
      WRITE(3,14) HEADER(1:17)
      WRITE(3,2)
      WRITE(3,13) WORDS
      CLOSE(2)
```

```
      GO TO 50
101   WRITE(3,4)
      GO TO 50
104   WRITE(3,10) FILNAM(1:J1)
      CLOSE(2)
      GO TO 50
105   WRITE(3,11) FILNAM(1:J1)
      CLOSE(2)
      GO TO 50
102   WRITE(6,5)
      STOP
      END
```

5.4 Flowchart for WHATSON



Prior to executing the FORTRAN program, the time of day and the directory of disk file names are read into a temporary file, SRF:WHATSON.DAT.

The time of day and the name of the disk are read from SRF:WHATSON.DAT

Read a file name from SRF:WHATSON.DAT and check to make sure that it is a data file.

Add punctuation and strip out the blanks so that the file name can be read correctly and opened as a logical unit.

Open the data file as a logical unit and read the first ten lines.

Print the file name and the first ten lines, then go back to line 37 to read the name of the next data file.

Statements for handling errors and the encounter with an end of file.

5.5 Example of WHATSON Printout

```
9/27/82   12:54:38
HERE"S WHAT"S ON  DISK SD8 : TODAY.


                                                             SD8  :A07SEP82.DAT

SD8:A07SEP82.DAT
MCDONNELL DOUGLAS ASTRONAUTICS CO.; S/N EI24548;
TALYSURF; 2,000X VERTICAL; 4X HORIZONTAL; 0.75 MM CUTOFF
    0.9500
   12.7300
 0.648671E-02
11
 1     3.0520
   85  113  141  170  200  229  257  287  318  350  383  418  453  487  519  54
  573  599  625  651  676  702  729  754  778  800  816  824  824  815  801  78


                                                             SD8  :A24AUG82.DAT

SD8:A24AUG82.DAT
GAR ELECTROFORMING DIV.; SPECIMENS #8, 10, 12
TALYSURF; 10,000X VERT.; 4X HOR.; 7.5 MICRON RAD. STYLUS; 0.75 MM CUTOFF
    0.9500
    2.9930
 0.131316E-02
18
 1     0.2030
 -116   75  196  211  163  117   98  104  129  163  190  197  182  144   79   -
  -66  -55    0   44   44   28   35   66   95  104   96   81   65   67   96   14


                                                             SD8  :A11MAY82.DAT

SD8:A11MAY82.DAT
ULTRA SONIC REFERENCE BLOCKS; CS-2; URB-1
TALYSURF; 10,000X VERTICAL; 4X HORIZONTAL; 7.5 U-M RADIUS STYLUS
    0.9500
    2.9930
 0.132030E-02
14
 1     0.2679
  -82 -111 -145 -168 -169 -155 -138 -125 -115 -107  -99  -93  -93 -100 -114 -13
 -161 -198 -243 -292 -339 -385 -442 -518 -609 -706 -801 -871 -881 -823 -733 -64
```

6. AVRGRA

6.1 Summary

AVRGRA is used in calibrations to calculate the average values of the R_a and step height data gathered by ROUGHNES and STEPHGHT. The random uncertainty shown in the printout, sec. 6.5, represents three standard deviations but includes a statistical factor for finite samples. For a set of n values x_i the random uncertainty (RU) is given by

$$RU = [3/K(n)] \left(\sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n-1} \right)^{1/2} \tag{6.1}$$

where $\bar{x}$ is the average value and K(n) is the factor which depends on the number of values n. K(n) has been tabulated by Natrella [7] for a range of n values.

The calibration uncertainty is calculated from one of ten formulas, the choice of which depends on 1) whether R_a or step height is being calculated and 2) the calibration step height. Each formula depends explicitly on the measured step height. Similar formulas have been given previously [8], but a few changes have been made on them to yield the present ones. These changes result from ongoing improvements and remeasurements of the system including the introduction of a new calibration step, the remeasurement of two others, remeasurement of the system noise, and further analysis of the uncertainties of roughness measurement for a highly smooth surface. The total uncertainty (sec. 6.5) is the sum of the random and calibration uncertainties.

6.2 Operating System Commands

```
***** AVRGRA *****
*
LO .BG, SRF:AVRGRA.TSK
T .BG
CLOSE ALL
AS 3,L4:
AS 5,C:
AS 6,C:
ST
*
* RETURN TO "FASTMENU".
*
FASTMENU
$EXIT
```

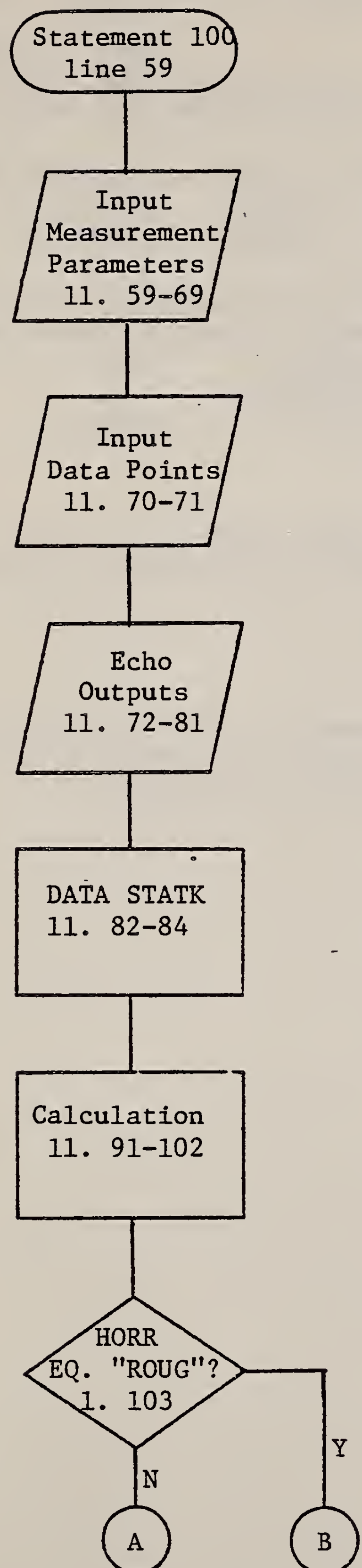
6.3 AVRGRA FORTRAN Program

```
C ***** SRF:AVRGRA.FTN   CALLED BY 'AVRGRA' *****
C THIS PROGRAM DOES TWO THINGS:
C 1) IT CALCULATES THE AVERAGE, AND STANDARD DEVIATION
C FOR A SET OF M DATA POINTS, WHICH ARE ENTERED AT THE
C CONSOLE. THESE DATA POINTS ARE NORMALLY THE RESULTS
C FOR ROUGHNESS AVERAGE (RA) OR STEP HEIGHT MEASUREMENTS
C OBTAINED IN THE "PROFILE" PROGRAM. NOTE THAT THE FACTOR
C FOR SMALL STATISTICAL SAMPLES IS INCLUDED IN THE STANDARD
C DEVIATION (LINES 80-82).
C 2) THE PROGRAM ALSO CALCULATES THE CALIBRATION UNCERTAINTY
C AND TOTAL UNCERTAINTY FOR RA AND STEP HEIGHT MEASUREMENTS.
C THEREFORE, IT MAINLY IS USED AS A SUMMARY IN CALIBRATION REPORTS.
C T. VORBURGER, 1978  (LAST REVISION 7/82)
C
C
      DIMENSION RA(25),STATK(25),DATE(10),SMPLID(20)
1     FORMAT(' SOMETHING"S WRONG. WHAT"S THE CALIBRATING STEP?')
   11 FORMAT(10X,8(2X,F10.5))
   12 FORMAT(F10.4)
   13 FORMAT(20A4)
   14 FORMAT(' NOW WE WILL CALCULATE THE AVERAGE ROUGHNESS',
     1' OR STEP HEIGHT OF'/' THE SURFACE, THE RANDOM',
     1' UNCERTAINTY, THE CALIBRATION'/' UNCERTAINTY, &',
     1' THE TOTAL UNCERTAINTY.'/' WHAT IS THE DATE OF THE',
     1' MEASUREMENTS?')
   15 FORMAT(' STEP OR ROUGHNESS?')
   16 FORMAT(' WHAT IS THE SAMPLE ID?')
   17 FORMAT(' WHAT IS THE HEIGHT OF THE CALIBRATING '
     1,'STEP IN UM?')
   18 FORMAT(' HOW MANY MEASUREMENTS? INCLUDE THE DECIMAL',
     1' POINT EXPLICITLY.')
   19 FORMAT(' NOW PUNCH IN THE',I3,' DATA POINTS IN',
     1' UM.'/' ONE ON EACH LINE.')
   20 FORMAT(10X,F7.4,' UM CALIBRATING STEP')
   21 FORMAT(/10X,'THE MEASURED POINTS IN UM ARE:')
   22 FORMAT(10X,5F10.4)
   23 FORMAT(//10X,'THE FINAL RESULTS IN UM ARE:'/
     118X,'RA',8X,'RANDOM   CALIBRATION',4X,'TOTAL'/
     124X,3(' UNCERTAINTY'))
   24 FORMAT(/10X,'THE FINAL RESULTS IN UM ARE:'/
     117X,'STEP',7X,'RANDOM   CALIBRATION',4X,'TOTAL'/
     116X,'HEIGHT',2X,,3(' UNCERTAINTY'))
   25 FORMAT(1H1///10X,20A4)
   26 FORMAT(10X,20A4)
   27 FORMAT(1X,20A4)
28    FORMAT(//' DO ANOTHER CALCULATION?'/' Y OR N?')
29    FORMAT(A2)
30    FORMAT(///' *** REMINDER! ***' /
     1' THE CALIBRATION UNCERTAINTY IS GIVEN FOR THE TALYSTEP.' /
     2' THE TALYSURF CU FOR A CALIBRATING STEP OF 0.301 UM' /
     3' MUST BE CALCULATED BY HAND.' ///)
C
C
C FIRST, THE OPERATOR IS ASKED TO TYPE IN SEVERAL
C PARAMETERS NEEDED IN THE CALCULATIONS.
C
C
      CALL CARCON (3,1)
```

```
  100 WRITE(6,14)
      READ(5,13)(DATE(I),I=1,10)
      WRITE(6,15)
      READ(5,13)HORR
      WRITE(6,16)
      READ(5,13)(SMPLID(I),I=1,20)
      WRITE(6,17)
      READ(5,12)CALSTP
      WRITE(6,18)
      READ(5,12)RM
      M = RM
      WRITE(6,19)M
      READ(5,12)(RA(I),I=1,M)
      WRITE(6,27)(DATE(I),I=1,10)
      WRITE(6,27)(SMPLID(I),I=1,20)
      WRITE(6,20)CALSTP
      WRITE(6,21)
      WRITE(6,22)(RA(I),I=1,M)
      WRITE(3,25)(DATE(I),I=1,10)
      WRITE(3,26)(SMPLID(I),I=1,20)
      WRITE(3,20)CALSTP
      WRITE(3,21)
      WRITE(3,22)(RA(I),I=1,M)
      DATA STATK/0.,.797,.886,.921,.940,.952,.959,.965,.969
     1,.973,.975,.977,.979,.981,.982,.983,.984,.985,.986
     2,.987,.987,.988,.988,.989,.989/
C
C
C THE CALCULATION OF THE AVERAGES AND STANDARD
C DEVIATIONS FOLLOWS NEXT.
C
C
      SUM = 0.
      DO 110 I=1,M
  110 SUM = SUM+RA(I)
      FINLRA = SUM/RM
      SUMDEV = 0.
      DO 112 I=1,M
      DEVSQ = (RA(I)-FINLRA)**2
  112 SUMDEV = SUMDEV+DEVSQ
      VAR = SUMDEV/(RM-1.)
      SD = SQRT(VAR)/STATK(M)
      RU = 3.*SD
      Q = FINLRA/CALSTP
      IF (HORR .EQ. 'ROUG') GO TO 130
C
C
C THE CALIBRATION UNCERTAINTY FOR STEP HEIGHTS IS
C CALCULATED IN THIS SECTION.
C
C
      WRITE (3,24)
      WRITE (6,24)
      IF (CALSTP .EQ. .025) GO TO 121
      IF (CALSTP .EQ. .301) GO TO 122
      IF (CALSTP .EQ. 2.993) GO TO 123
      IF (CALSTP .EQ. 12.73) GO TO 123
      IF (CALSTP .EQ. 22.9) GO TO 123
```

```
      WRITE (6,1)
      GO TO 200
121   CU = .001*(SQRT(6.1*Q**2 + 1.69) + 6.*Q)
      GO TO 140
122   CU = .001*(SQRT(233.*Q**2 + 4.) + 12.*Q)
      GO TO 140
123   CU = SQRT((.027*Q)**2 +(.01*FINLRA)**2 +(.01*CALSTP)**2)+.03*Q
      GO TO 140
C
C
C THE CALIBRATION UNCERTAINTY FOR ROUGHNESS IS
C CALCULATED IN THIS SECTION, AND THE FINAL RESULTS ARE PRINTED.
C
C
130   WRITE(3,23)
      WRITE(6,23)
      IF (CALSTP .EQ. .025) GO TO 131
      IF (CALSTP .EQ. .301) GO TO 132
      IF (CALSTP .EQ. 2.993) GO TO 133
      IF (CALSTP .EQ. 12.73) GO TO 133
      IF (CALSTP .EQ. 22.9) GO TO 133
      WRITE(6,1)
      GO TO 200
  131 CU = .001*(2.791*Q + SQRT(36.*Q*Q + .0881))
      GO TO 140
132   CU=.001*(SQRT(233.*Q**2+4.)+SQRT(144.*Q**2+16.0256))
      WRITE(6,30)
      GO TO 140
133   CU=SQRT((.027*Q)**2+.000296*FINLRA**2)+SQRT((.03*Q)**2+3.6E-5)
140   TU = RU+CU
      WRITE(3,11) FINLRA,RU,CU,TU
      WRITE(6,11) FINLRA,RU,CU,TU
200   WRITE (6,28)
      READ (6,29) QUERY
      IF (QUERY .EQ. 'Y ') GO TO 100
      STOP
      END
```

6.4 Flowchart for AVRGRA



The operator is prompted to type the date, the choice of whether these will be R_a or step data, the sample I.D., the calibrating step height, and the number of data points.

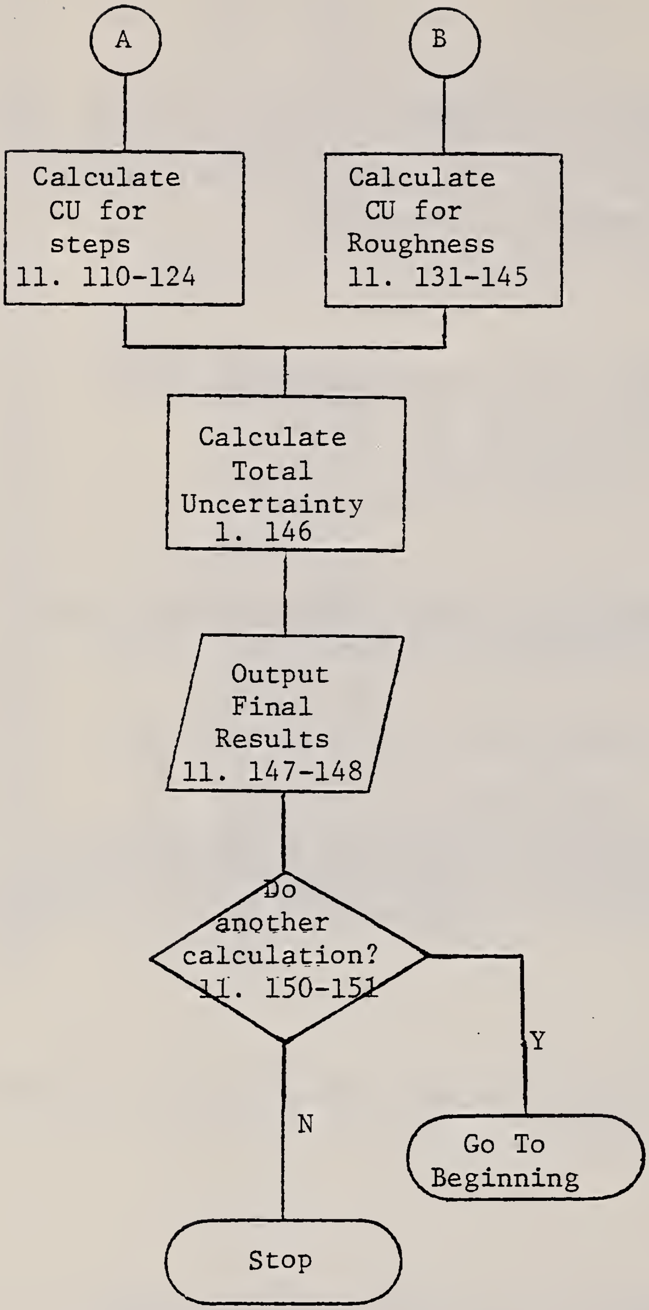
The operator now types in the data points.

The above data are now printed on the printer and the console.

The statistical factors which figure in the calculation of standard deviation for finite data sets [7] are entered in this data statement.

Calculation of averages and standard deviations.

The branch to the calibration uncertainty calculation depends on whether these are step heights or roughness data.



Calculation of Calibration Uncertainty.

Total Uncertainty
= Random + Calibration Uncertainties

The average R_a or step height and the uncertainties are printed on the console and the printer.

6.5 Example of AVRGRA Printout

```
JUNE 5, 1980
0.3 UM PROTOTYPE SRM
 2.9930 UM CALIBRATING STEP

THE MEASURED POINTS IN UM ARE:
       0.2953     0.2949     0.2949     0.2944     0.2946
       0.2950     0.2942     0.2950     0.2946     0.2928


THE FINAL RESULTS IN UM ARE:
            RA          RANDOM    CALIBRATION     TOTAL
                     UNCERTAINTY UNCERTAINTY UNCERTAINTY
         0.29457       0.00216      0.01241      0.01457
```

7. SMORGAS

7.1 Summary

This program calculates seven surface texture parameters from profile data. These are the roughness average - R_a, rms roughness - R_q, peak-to-valley roughness - R_{tm}, average slope - S_a, average wavelength - D_a, a new parameter we call the peak-count wavelength - D_{pc}, and skewness - Q. The parameters are calculated for each profile in a file as well as the averages and standard deviations of the mean for the set of profiles. The key elements in the program are the formulas used to calculate the various parameters. They are summarized as follows:

$$R_a = (1/N) \sum_{i=1}^{N} |y_i|, \tag{7.1}$$

where the y_i's represent the set of digitized profile ordinates measured with respect to the mean line. The total number of points (N) in the profile is equal to 4000.

$$R_q = \left[(1/N) \sum_{i=1}^{N} y_i^2 \right]^{1/2} . \tag{7.2}$$

$$Q = (1/N) \sum_{i=1}^{N} y_i^3/R_q^3. \tag{7.3}$$

$$R_{tm} = (1/J) \sum_{j=1}^{J} (y_{max}-y_{min})_j, \tag{7.4}$$

where $(y_{max}-y_{min})j$ is the height difference between the highest peak and lowest valley in a given sampling length, a quantity chosen by the operator. The sampling length is divided into the total profile length to give the integral number of sampling lengths J.

$$S_a = (1/PkT) \sum_{j=1,1+k,1+2k...}^{1+Pk} |y_{j+k}-y_j|, \tag{7.5}$$

where T is the horizontal point spacing of the digitized profile and kT is the horizontal point spacing, chosen by the operator, that determines the resolution of the calculation. The quantity

kT is equal to an integral number of point spacings in the profile itself and P is the total number of these kT spacings in the profile.

The wavelength parameters, D_a and D_{pc}, attempt to quantify the average horizontal distance between adjacent irregularities of a surface profile. Since, for a random profile, this involves judgement as to the definition of the irregularities themselves, the parameters, D_a and D_{pc}, calculate the wavelength in two different ways.

The average wavelength, D_a, uses the ratio of the height parameter R_a to the average slope [9],

$$D_a = 2\pi R_a/S_a. \tag{7.6}$$

Alternatively, the peak-count wavelength D_{pc} relates the irregularities to a definition of the significant peaks and valleys of the profile.

$$D_{pc} = 2(N-1)T/X, \tag{7.7}$$

where (N-1)T is the total profile length and X is the number of times that the profile crosses completely through a mean band with height equal to R_a, that is centered around the profile mean line [10]. If the profile were a perfect sine wave, D_{pc} and D_a would yield essentially the same value of wavelength.

There are three important options which the operator must provide SMORGAS:

1. the sampling length for the R_{tm} calculation,
2. the point to point spacing kT for the average slope calculation, and
3. the choice of calculating a mean line by taking the least squares straight line fit to the data or by taking the simple mean of the data points.

7.2 Operating System Commands

```
 1        ***** SMORGAS *****
 2        *
 3        LO .BG, SRF:SMORGAS.TSK
 4        T.BG;  CLOSE ALL
 5        AS 3,L4:;  AS 6,C:
 6        $IFNN @1
 7          $IFNX @1
 8            $W*;  $W*
 9            $W* THE INPUT FILE YOU NAMED DOES NOT EXIST.
10            $W* TRY AGAIN .
11            $W*;  $W*
12            CA .BG
13            $EXIT
14          $ENDC
15          AS 5,@1
16        $ENDC
17        $IFNU @1
18          AS 5,C:
19        $ENDC
20        ST
21        *
22        * RETURN TO "FASTMENU".
23        *
24        FASTMENU
25        $EXIT
```

Notes:
The statements on lines 6-18 direct the computer to read the input parameters either from a data file or from the system console.

7.3 SMORGAS FORTRAN Program

```
C ***** SRF:SMORGAS.FTN CALLED BY THE COMMAND 'SMORGAS' *****
C THIS PROGRAM CALCULATES A SMORGASBORD OF SURFACE
C ROUGHNESS PARAMETERS FOR SURFACE PROFILES. THESE ARE
C ROUGHNESS AVERAGE-RA, RMS ROUGHNESS-RQ, PEAK-VALLEY ROUGHNESS-RTM,
C AVERAGE SLOPE-SA, AVERAGE WAVELENGTH-DA,
C PEAK-COUNT WAVELENGTH-DPC, AND SKEWNESS.
C THE STANDARD DEVIATION CALCULATED AT THE END
C INCLUDES THE STATISTICAL K FACTOR FOR FINITE
C SAMPLES.
C      T.VORBURGER 3/79 (REVISED 2/26/82)
C
C
       DIMENSION RA(75),RQ(75),SA(75),RTM(75),SKEW(75)
       DIMENSION DPC(75),PC(75)
       INTEGER SUM1,SPEC(20),NRUN(75),SUMRA,SUMSA,CUTOFF,UP2,LOW2
       INTEGER*2 PROFIL(4000),DATFIL(9),A,B
       REAL KCAL,NSUM,LAMBDA(75)
       DATA PI,DATFIL(7),DATFIL(8),DATFIL(9)/
      13.14159,2H.D,2HAT,2H  /
C
C
1      FORMAT(F10.4)
2      FORMAT(40A2)
3      FORMAT(20A4)
4      FORMAT(' PRESENTING A SMORGASBORD OF SURFACE'/
      1' PARAMETERS FOR YOUR ANALYTICAL PLEASURE!'/
      2' WHAT IS THE FILE NAME? FORMAT SHOULD'/
      3' BE "VOL:NAME" WITH EXACTLY 12 CHARACTERS.')
5      FORMAT(I2)
6      FORMAT(/' POINT TO POINT SPACING =',F8.4,' UM')
7      FORMAT(/' KCAL =',E13.6, ' UM/QUANTIZATION LEVEL')
8      FORMAT(/' THERE ARE ', I2,' PROFILES')
9      FORMAT(16I5)
10     FORMAT(1X,I2,3F10.4,E13.4,F9.3,F6.1,2F9.3)
11     FORMAT(' ERROR STATUS = ',I2,' ON OPENW ROUTINE'//
      1' COME ON, TURKEY. DO IT OVER AND GET IT RIGHT THIS TIME!')
12     FORMAT(1X,I2,3F10.5,E13.4,F9.3,F6.1,2F9.3)
13     FORMAT(3X,3F10.5,E13.4,F9.3,6X,2F9.3)
14     FORMAT(//' DO YOU WISH TO CONTINUE?'//' Y OR N?')
15     FORMAT(///1X,' POS',4X,'RA',8X,'RQ',7X,'RTM',9X,'AVE',
      17X,'AVE',6X,'PC',1X,'PEAK-COUNT',3X,'SKEW'/
      239X,'SLOPE',2X,'WAVELENGTH',6X,'WAVELENGTH'/)
16     FORMAT(/10X,'AVERAGES WITH THREE STANDARD DEVIATIONS',
      1' OF THE MEAN (UM)'/)
17     FORMAT(3X,3F10.4,E13.4,F9.3,6X,2F9.3)
18     FORMAT(E13.6)
19     FORMAT(//' THE DATA ARE BEING FITTED TO A LEAST SQUARES'/
      1' STRAIGHT LINE FOR THIS CALCULATION.')
20     FORMAT(' WHAT IS THE POINT-TO-POINT SPACING'/
      1' FOR THE SLOPE CALCULATION? A GOOD CHOICE'/
      2' IS THE STYLUS RADIUS OR, IF THE STYLUS IS FLAT'/
      3' ON TOP, THE WIDTH OF THE STYLUS TIP.')
21     FORMAT(/ ' EXPRESS THE VALUE IN UM & INCLUDE'/
      5' THE DECIMAL POINT EXPLICITLY.')
22     FORMAT(' SHOULD WE DO A LEAST SQUARES STRAIGHT LINE FIT'/
      1' TO THE DATA?  "Y" OR "N"?')
23     FORMAT(' I CAN"T READ THIS NAME. TRY AGAIN.')
24     FORMAT(1H ,40A2)
```

```
 25    FORMAT(1H1)
 26    FORMAT(' WE ROUNDED OFF THE VALUE YOU SUGGESTED'/
      1' TO',F10.4,' SO THAT THE POINT-TO-POINT SPACING'/
      2' FOR THE SLOPE CALCULATION IS EXACTLY ',I2/
      3' TIMES THE POINT-TO-POINT SPACING OF THE DATA.')
 27    FORMAT(/F8.4,' UM = POINT-TO-POINT SPACING',
      1' FOR THE SLOPE CALCULATION.'/' THIS = ',I3,
      2' TIMES THE DATA POINT SPACING.')
 28    FORMAT(' WHAT IS THE SAMPLING LENGTH FOR THE RTM CALCULATION?'/
      1' A GOOD CHOICE IS 800 UM.')
 29    FORMAT(1X,20A4)
 30    FORMAT(/F9.3,' UM = SAMPLING LENGTH FOR THE RTM CALCULATION.'/
      1' THIS = ',I4,' TIMES THE DATA POINT SPACING.')
 31    FORMAT(//' A MEAN VALUE IS BEING SUBTRACTED FROM THE DATA.')
C
C FIRST, THE OPERATOR IS ASKED TO TYPE THE NAME OF THE DATA
C FILE. THIS FILE IS THEN ASSIGNED TO LU 10 AND SUBSEQUENT
C DATA ARE READ FROM IT. THE OPERATOR IS ALSO PROMPTED TO
C TYPE THE INTERVAL SPACING FOR THE SLOPE CALCULATION
C AND THE SAMPLE INTERVAL FOR RTM.
C
 95     ITRY = 0
 92     WRITE(6,4)
        READ(5,2,ERR=93,END=703)(DATFIL(J),J=1,6)
        WRITE(6,24)DATFIL
        GO TO 94
 93     ITRY = ITRY + 1
        IF (ITRY .GT. 2) GO TO 703
        WRITE(6,23)
        GO TO 92
 94     CALL OPENW(10,DATFIL,0,0,0,ISTAT)
        IF(ISTAT.LT.1)GO TO 90
        WRITE(6,11)ISTAT
        ITRY = ITRY + 1
        IF (ITRY .GT. 2) GO TO 703
        GO TO 92
 90     READ(10,3,REC=2)SPEC
        READ(10,1,REC=4)PTTOPT
        READ(10,18,REC=6)KCAL
        READ(10,5)NTOT
        WRITE (3,25)
        WRITE(3,24)DATFIL
        WRITE(3,29)SPEC
        WRITE(3,6)PTTOPT
        WRITE(3,7)KCAL
        WRITE(3,8)NTOT
        WRITE (6,28)
        WRITE (6,21)
        READ (5,1) SAMLGT
        RCUT = SAMLGT/PTTOPT
        CUTOFF = NINT(RCUT)
        IF (CUTOFF .GT. 4000) CUTOFF = 4000
        SAMLGT = CUTOFF*PTTOPT
        WRITE (6,30) SAMLGT,CUTOFF
        WRITE (3,30) SAMLGT,CUTOFF
        WRITE(6,20)
        WRITE (6,21)
        READ(5,1) RPT
```

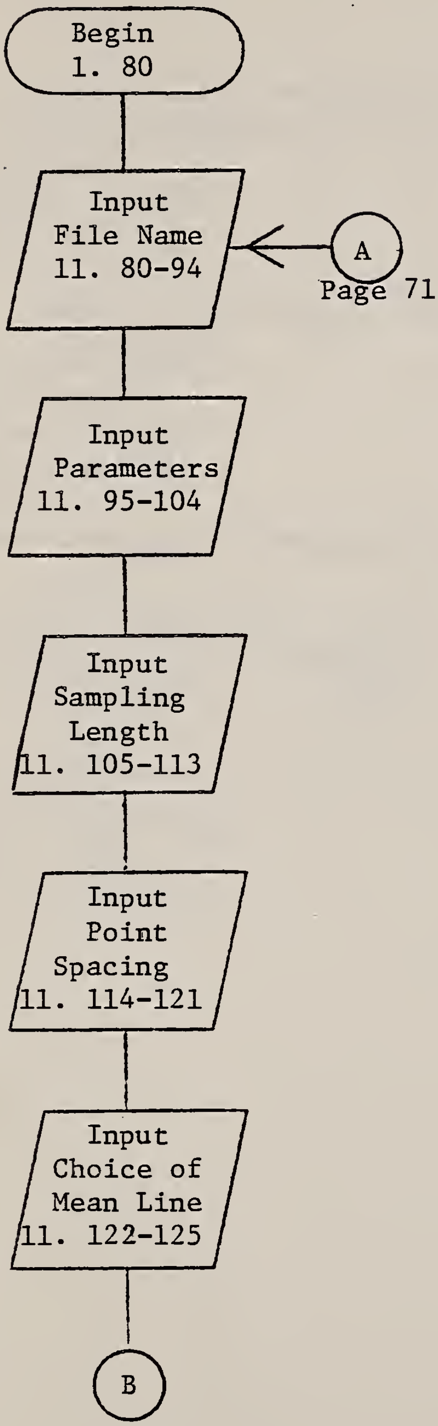
```
      RSPACE = RPT/PTTOPT
      NSPACE = NINT(RSPACE)
      RPT = NSPACE*PTTOPT
      WRITE(6,26) RPT,NSPACE
      WRITE(3,27) RPT,NSPACE
      WRITE (6,22)
      READ (5,2) QUERY1
      IF (QUERY1 .EQ. 'Y ') WRITE(3,19)
      IF (QUERY1 .EQ. 'N ') WRITE(3,31)
C
C NOW, WE START THE MAIN LOOP IN WHICH EACH SET OF
C PROFILE DATA IS READ AND ANALYZED TO FIND THE ABOVE QUANTITIES
C
      DO 900 IRUN=1,NTOT
        READ(10,5)NRUN(IRUN)
        READ(10,9)PROFIL
C
C AT THIS POINT, WE SUBTRACT A MEAN VALUE
C FROM THE PROFILE DATA OR DO A LEAST SQUARES
C STRAIGHT LINE FIT.
C
        IF (QUERY1 .EQ. 'Y') GO TO 210
        CALL MEAN(PROFIL,4000)
        GO TO 220
210     CALL LEASQ(PROFIL,4000)
C
C NOW, WE CALCULATE RA, RQ, AND SKEWNESS.
C
220     SUMRA = 0
        SUMRQ = 0.
        SUMSKW = 0.
        DO 200 I = 1,4000
          SUMRA = SUMRA + ABS(PROFIL(I))
          SUMRQ = SUMRQ + PROFIL(I)*PROFIL(I)
          RPRO = PROFIL(I)
          SUMSKW = SUMSKW + RPRO*RPRO*RPRO
200     CONTINUE
        RA(IRUN) = SUMRA*KCAL/4000.
        RQ(IRUN) = SQRT(SUMRQ/4000.)*KCAL
        SKEW(IRUN) = SUMSKW*SQRT(4000.)/(SUMRQ**1.5)
C
C AT THIS POINT, WE CALCULATE RTM.
C
        PVSUM = 0.
        NUMCUT = 4000/CUTOFF
        DO 800 I = 1,NUMCUT
          UP2 = -2048
          LOW2 = 2048
          DO 801 J=1,CUTOFF
            K = (I-1)*CUTOFF + J
            IF (PROFIL(K) .GT. UP2) UP2 = PROFIL(K)
            IF (PROFIL(K) .LT. LOW2) LOW2 = PROFIL(K)
801       CONTINUE
       IWRITE = UP2-LOW2
800     PVSUM = UP2 - LOW2 + PVSUM
        RCUT = NUMCUT
        RTM(IRUN) = KCAL*PVSUM/RCUT
C
```

```
C NOW, WE CALCULATE THE AVERAGE SLOPE,
C AND THE AVERAGE WAVELENGTH.
C
          SUMSA = 0
          NSUM = 0.
          DO 300 I=1,3999,NSPACE
            IUP = I + NSPACE
            IF (IUP .GT. 4000) GO TO 300
            DIFF=PROFIL(IUP)-PROFIL(I)
            SUMSA = SUMSA + ABS(DIFF)
            NSUM = NSUM + 1.
300         CONTINUE
C
          SA(IRUN) = SUMSA*KCAL/(NSUM*RPT)
          LAMBDA(IRUN) = 2.*PI*RA(IRUN)/SA(IRUN)
C
C NOW WE COMPUTE THE PEAK-COUNT WAVELENGTH. IT IS DEFINED
C AS THE PROFILE LENGTH DIVIDED BY HALF THE NUMBER OF TIMES
C THAT THE PROFILE CROSSES A MEAN BAND CENTERED ABOUT THE
C MEAN LINE. THE WIDTH OF THE MEAN BAND IS EQUAL TO RA.
C
          UP = SUMRA/8000.
          RLOW = -SUMRA/8000.
C
C NOW, WE DETECT AND COUNT THE BANDWIDTH CROSSINGS.
C
          A=0
          B=0
          E=0.0
          DO 500 I=1,4000
            IF(PROFIL(I).EQ.UP.OR.PROFIL(I).EQ.RLOW)GO TO 500
410         IF(PROFIL(I).LT.UP)GO TO 420
            IF(A.EQ.1)GO TO 500
            A=1
            GO TO 450
420         IF(PROFIL(I).LT.RLOW)GO TO 430
            GO TO 450
430         IF(B.EQ.1)GO TO 500
            B=1
450         IF((A+B).NE.2)GO TO 500
            E=E+1.0
            A=0
            B=0
            GO TO 410
500       CONTINUE
          PC(IRUN) = E/2.
          DPC(IRUN) = 7998.*PTTOPT/E
900     CONTINUE
C
C THE RESULTS FOR EACH PROFILE ARE NOW PRINTED
C
        WRITE(3,15)
        DO 600 K=1,NTOT
          IF (RA(K).LT.0.001) GO TO 601
600     CONTINUE
        GO TO 602
601     WRITE(3,12) (NRUN(K),RA(K),RQ(K),RTM(K),SA(K),
       1LAMBDA(K),PC(K),DPC(K),SKEW(K),K=1,NTOT)
```

```
      GO TO 603
602   WRITE(3,10) (NRUN(K),RA(K),RQ(K),RTM(K),SA(K),
     1LAMBDA(K),PC(K),DPC(K),SKEW(K),K=1,NTOT)
C
C THE MAIN LOOP IS COMPLETE.
C FINALLY WE CALCULATE THE AVERAGES AND STANDARD
C DEVIATIONS OF ALL THE ABOVE QUANTITIES AND PRINT THEM.
C
603   CALL AVSD(NTOT,RA,FNLRA,SDRA)
      CALL AVSD(NTOT,RQ,FNLRQ,SDRQ)
      CALL AVSD(NTOT,RTM,FNLRTM,SDRTM)
      CALL AVSD(NTOT,SA,FNLSA,SDSA)
      CALL AVSD(NTOT,LAMBDA,FNLLAM,SDLAM)
      CALL AVSD(NTOT,DPC,FNLDPC,SDDPC)
      CALL AVSD(NTOT,SKEW,FNLSKW,SDSKW)
      WRITE (3,16)
      CALL CLOSE(10,ISTATE)
      IF (FNLRA .LT. 0.001) GO TO 701
      GO TO 702
701   WRITE(3,13) FNLRA,FNLRQ,FNLRTM,FNLSA,FNLLAM,
     1FNLDPC,FNLSKW,SDRA,SDRQ,SDRTM,SDSA,
     2SDLAM,SDDPC,SDSKW
      GO TO 92
702   WRITE (3,17) FNLRA,FNLRQ,FNLRTM,FNLSA,FNLLAM,
     1FNLDPC,FNLSKW,SDRA,SDRQ,SDRTM,SDSA,
     2SDLAM,SDDPC,SDSKW
      WRITE (6,14)
      READ (5,2) QUERY2
      IF (QUERY2 .EQ. 'Y ') GO TO 95
703   STOP
      END
C
C
C THIS SUBROUTINE CALCULATES THE MEAN AND STAN-
C DARD DEVIATION OF THE MEAN OF A SET OF DATA VALUES.
C
C
      SUBROUTINE AVSD(N,VALUE,FNL,SDM)
      DIMENSION VALUE(75),STATK(25)
      DATA STATK/0.,.797,.886,.921,.940,.952,.959,.965,.969
     1,.973,.975,.977,.979,.981,.982,.983,.984,.985,.986
     2,.987,.987,.988,.988,.989,.989/
      SUM1 = 0.
      RN = N
      SUM2 = 0.
      DO 100 I = 1,N
        SUM1 = SUM1 + VALUE(I)
100   CONTINUE
      FNL = SUM1/RN
      DO 200 I = 1,N
        DEV = VALUE(I) - FNL
        SUM2 = SUM2 + DEV**2
200   CONTINUE
      SDM = 3.*SQRT(SUM2/(RN*(RN-1.)))/STATK(N)
      RETURN
      END
C
C NEXT, WE CALCULATE THE MEAN OF THE DATA AND SUBTRACT
```

```
C IT FROM EACH POINT OF THE SURFACE PROFILE.
C
      SUBROUTINE MEAN(PROFIL,N)
      INTEGER*2 PROFIL(N)
      INTEGER SUM1
        SUM1=0
        DO 50 I=1,4000
          SUM1=SUM1+PROFIL(I)
50      CONTINUE
        RMEAN = SUM1/4000.
        MEEN = NINT(RMEAN)
        DO 100 I=1,4000
          PROFIL(I) = PROFIL(I) - MEEN
100     CONTINUE
      RETURN
      END
C
C
C THIS ROUTINE CALCULATES THE LEAST SQUARES STRAIGHT LINE FOR
C A SET OF 4000 EQUALLY SPACED DATA POINTS. IT CAN BE USED TO
C FILTER OUT ANY SLOPE IN A SET OF PROFILE DATA.
C
      SUBROUTINE LEASQ(PROFIL,N)
      INTEGER*2 PROFIL(N)
      DOUBLE PRECISION SUMX2,SUMXY,DELTA,AZERO,AONE
      RN = N
C
C IN THE LEAST SQUARES FIT, WE HAVE ALREADY
C CALCULATED THE SUM OF THE X'S AND X SQUARES,
C WHOSE VALUES NEVER CHANGE. THIS AVOIDS HAVING
C TO CALCULATE THEIR VALUES IN THE LOOP.
C
      SUMX1 = 8002000.
      SUMX2 = 2.1341334D10
      SUMY1 = 0.
      SUMXY = 0.
      DO 101 I=1,N
        SUMY1 = SUMY1 + PROFIL(I)
        SUMXY = SUMXY + I*PROFIL(I)
101   CONTINUE
C
      DELTA = RN*SUMX2 - SUMX1*SUMX1
      AZERO = (SUMX2*SUMY1 - SUMX1*SUMXY)/DELTA
      AONE = (RN*SUMXY - SUMX1*SUMY1)/DELTA
C
      DO 102 I=1,N
        RI = I
        SUB = AZERO + AONE*RI
        X = 0.5
        IF (SUB .LT. 0.) X = -X
        NSUB = SUB + X
        PROFIL(I) = PROFIL(I) - NSUB
102   CONTINUE
      RETURN
      END
```

7.4 Flowchart for SMORGAS



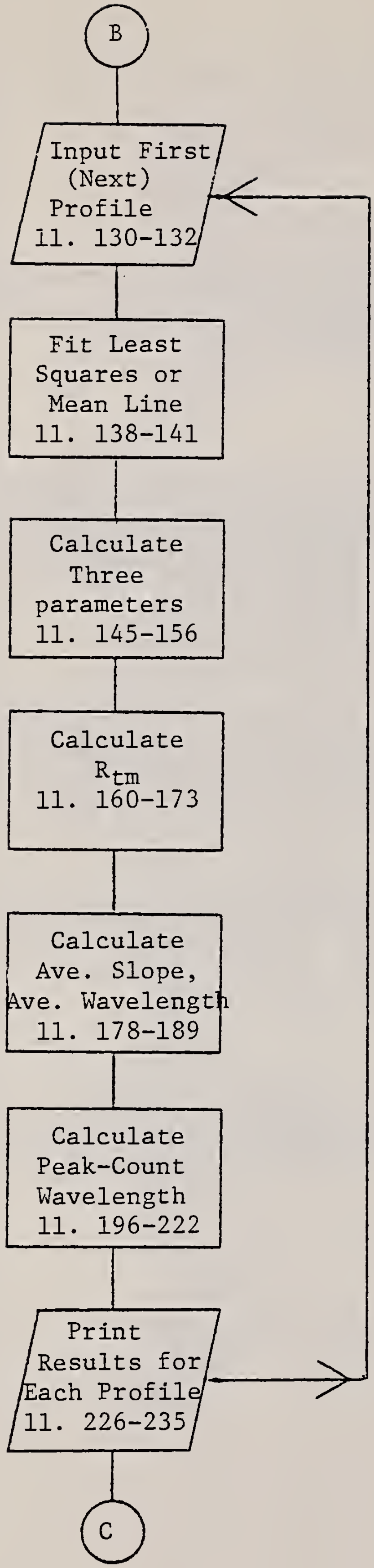
The operator is prompted to type the name of the data file to be examined. The file is then opened as a logical unit.

The program reads the specimen I.D., the horizontal spacing of the data points, the instrument calibration constant, and the total number of profiles in the file. The information is then printed out.

The operator is prompted to input the sampling length for the R_{tm} calculation.

The operator is prompted to input the point-to-point spacing for the slope calculation.

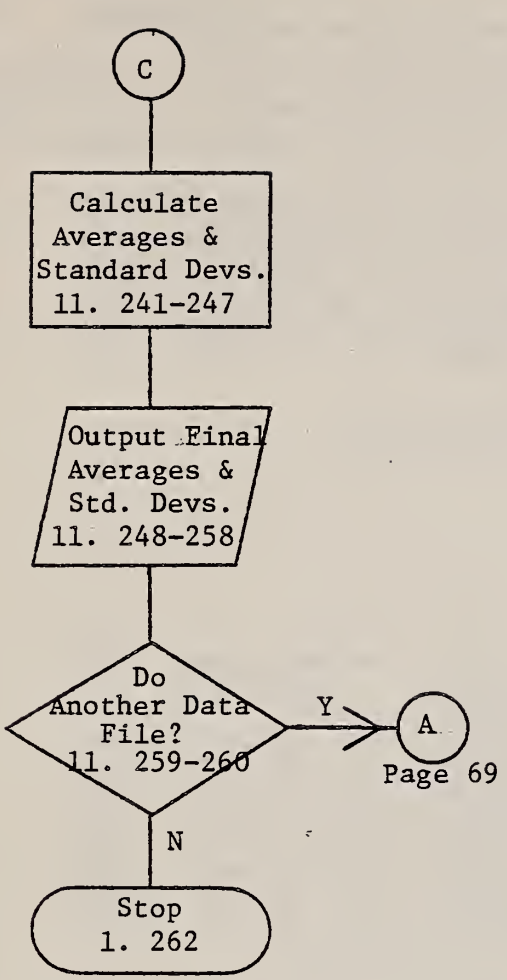
The operator is prompted to choose whether or not to fit a least squares line to the data to be used as the mean line.



The loop to calculate the parameters for each profile begins here.

A subroutine, either LEASQ or MEAN, is called.

R_a, R_q, and the skewness are calculated.


C
Calculate
Averages &
Standard Devs.
11. 241-247
Output Final
Averages &
Std. Devs.
11. 248-258
Do
Another Data
File?
11. 259-260
Y
A
Page 69
N
Stop
1. 262

Call AVSD.

7.5 Example of SMORGAS Printout

The surface described here is a prototype calibration specimen that was specially machined to have a sinusoidal profile [11].

```
 3:SRMS3MUM.DAT
SINUSOIDAL PROTOTYPES WITH 3 UM RA, 28APR78

POINT TO POINT SPACING =  0.9500 UM

KCAL = 0.672072E-02 UM/QUANTIZATION LEVEL

THERE ARE  8 PROFILES

 799.900 UM = SAMPLING LENGTH FOR THE RTM CALCULATION.
THIS =  842 TIMES THE DATA POINT SPACING.

 7.6000 UM = POINT-TO-POINT SPACING FOR THE SLOPE CALCULATION.
THIS =   8 TIMES THE DATA POINT SPACING.


THE DATA ARE BEING FITTED TO A LEAST SQUARES
STRAIGHT LINE FOR THIS CALCULATION.
```

POS	RA	RQ	RTM	AVE SLOPE	AVE WAVELENGTH	PC	PEAK-COUNT WAVELENGTH	SKEW
1	2.9611	3.3075	9.7921	0.1865E+00	99.749	37.5	101.308	-0.093
2	2.9737	3.3164	9.6644	0.1857E+00	100.623	37.0	102.677	-0.114
3	2.9786	3.3171	9.6896	0.1859E+00	100.676	37.5	101.308	-0.091
4	2.9675	3.3096	9.7417	0.1861E+00	100.174	37.5	101.308	-0.110
5	2.9388	3.2879	9.7299	0.1863E+00	99.115	37.5	101.308	-0.112
6	3.0074	3.3455	9.7098	0.1861E+00	101.539	37.5	101.308	-0.091
7	2.9701	3.3107	9.7047	0.1861E+00	100.289	37.5	101.308	-0.084
8	3.0121	3.3441	9.6476	0.1860E+00	101.757	37.5	101.308	-0.089

AVERAGES WITH THREE STANDARD DEVIATIONS OF THE MEAN (UM)

2.9762	3.3173	9.7100	0.1861E+00	100.490		101.479	-0.098
0.0263	0.0211	0.0501	0.2761E-03	0.960		0.532	0.013

8. PLOTSVIL

8.1 Summary

PLOTSVIL provides plots of surface profiles. The operator first chooses the name of the file that contains the desired data. Then the operator has the option of choosing,

1. the number of profiles to plot and which ones to plot,
2. the beginning and ending data points to be plotted (ifthe entire profile is to be plotted, the beginning point is 1 and the ending point is 4000), and
3. the length of the plot in cm.

8.2 Operating Systems Commands

```
***** PLOTSVIL *****
*
VPHS1 SRF:PLOTSVIL,@1
VPHS2 SRF:PLOTSVIL,3
$W*
$W*
$W*
$W* THE PLOTTING PROGRAM HAS BEEN COMPLETED.
$W* THE RASTER FILE "S:RASDATA.RAS" HAS BEEN CREATED
$W* AND MAY BE PLOTTED ON THE 3230 SYSTEM USING THE COMMAND "PLOTFINI
".
$W*
$W* TO LOOK AT THE ENTIRE SET OF COMMANDS FOR
$W* MEASURING AND ANALYSING SURFACE PROFILE DATA,
$W* USE THE COMMAND "FASTMENU".
$EXIT
```

Notes:
"VPHS1" and "VPHS2" are sets of operating system commands used when programs that plot graphs are executed.

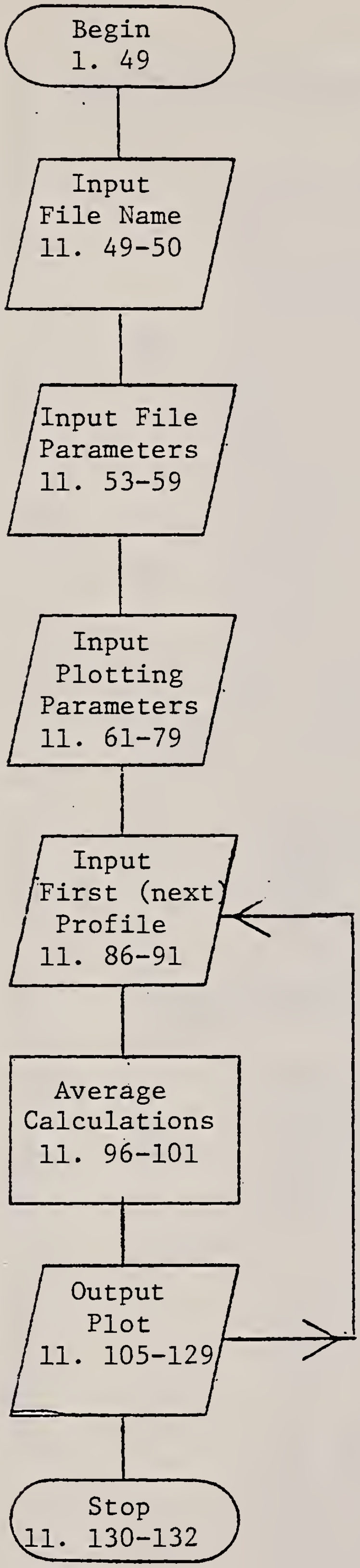
8.3 PLOTSVIL FORTRAN Program

```
C ***** SRF:PLOTSVIL.FTN  CALLED BY "PLOTSVIL" *****
C THIS ROUTINE IS USED TOGETHER WITH VERSATEC PLOTTING SOFTWARE
C TO PRODUCE PLOTS OF THE DIGITIZED SURFACE PROFILES.
C THE OPERATOR IS PROMPTED TO TYPE THE FILE NAME
C WHERE THE DATA CAN BE FOUND, THE NUMBERS OF THE
C PROFILES THAT ARE TO BE PLOTTED, THE NUMBER OF
C POINTS TO BE PLOTTED IN EACH ONE, AND THE
C LENGTH OF EACH PLOT ON THE OUTPUT PAGE.
C       T. VORBURGER, 1978 (LAST REVISION 3/82)
C
C
      DIMENSION X(4002),Y(4002),SPEC(20)
      INTEGER*2 DATFIL(9),PRONUM(50),START
      REAL KCAL,LENGTH
      DATA DATFIL(7),DATFIL(8),
     1DATFIL(9)/'.D','AT','  '/
C
1     FORMAT(F10.4)
2     FORMAT(16F5.0)
3     FORMAT(' WHAT IS THE DATA FILE? 12 CHARACTERS')
4     FORMAT(E13.6)
5     FORMAT(40A2)
6     FORMAT(I2)
7     FORMAT(' THERE ARE A TOTAL OF',1X,I2,1X,'PROFILES.'/
     +' HOW MANY PROFILES DO YOU WANT GRAPHED? I2 FORMAT')
8     FORMAT(' INPUT THE NUMBER OF THE NEXT PROFILE DESIRED.'/
     1' USE I2 FORMAT. THEN HIT RETURN.')
9     FORMAT(1X,10F6.0)
10    FORMAT(' INPUT THE NUMBER OF THE FIRST PROFILE DESIRED.'/
     1' USE I2 FORMAT. THEN HIT RETURN.')
11    FORMAT(' THE AVERGE HEIGHT FOR PROFILE',1X,I2,1X,'IS',
     +1X,F10.4,1X,'MICRONS')
12    FORMAT(20A4)
13    FORMAT(' AT WHAT POINT ON THE PROFILE WOULD YOU LIKE',
     1' TO START THE PLOT?'/' ANSWER CAN BE BETWEEN 1 AND',
     2' 4000 IN I4 FORMAT.')
14    FORMAT(I4)
15    FORMAT(' HOW MANY POINTS DO YOU WANT TO PLOT?',
     1' I4 FORMAT AGAIN.')
16    FORMAT(' HOW LONG SHOULD THE PLOT BE IN CM?',
     1' USE I3 FORMAT.')
17    FORMAT(I3)
18    FORMAT(' THERE ARE ONLY 4000 POINTS IN THE WHOLE',
     1' PROFILE, DUMMY. TRY AGAIN.')
C
C FIRST, THE OPERATOR TYPES IN THE DATA FILE NAME
C AND THE NECESSARY PARAMETERS.
C
      WRITE(6,3)
      READ(5,5) (DATFIL(J),J=1,6)
C
C
      CALL OPENW(9,DATFIL,0,0,0,STATUS)
      READ(9,12,REC=2)SPEC
      READ(9,1,REC=4)PTTOPT
      WRITE(6,1)PTTOPT
      READ(9,4,REC=6)KCAL
      WRITE(6,1)KCAL
```

```
      READ(9,6)ITOTAL
C
C
      WRITE(6,7)ITOTAL
      READ(5,6)ITOPRO
      WRITE(6,10)
      READ(5,6) PRONUM(1)
      DO 50 M=2,ITOPRO
        WRITE(6,8)
50    READ(5,6)PRONUM(M)
65    WRITE(6,13)
      READ(5,14) N0
      WRITE(6,15)
      READ(5,14) N1
      NCHECK = N1+N0
      IF (NCHECK.LE.4001) GO TO 60
      WRITE(6,18)
      GO TO 65
60    WRITE(6,16)
      READ(5,17) L
      LENGTH =  L/2
      CALL PLOTS(0,0,0)
      CALL FACTOR(.7874)
      CALL PLOT(1.,3.,-3)
C
C THEN, THE PROGRAM READS THE RELEVANT PROFILE DATA.
C
      DO 300 M=1,ITOPRO
        RNUMBR=PRONUM(M)
        START=(PRONUM(M)-1)*251+8
        READ(9,6,REC=START)ITEST
        READ(9,2)(Y(I),I=1,4000)
        WRITE(6,9)(Y(I),I=1,10)
C
C THE AVERAGE MEAN HEIGHT OF THE PROFILE DATA IS
C CALCULATED HERE AS A CHECK.
C
        SUM=0.0
        DO 100 N=1,4000
          SUM=SUM+Y(N)
100     CONTINUE
        AVE=KCAL*SUM/4000.0
        WRITE(3,11)PRONUM(M),AVE
C
C FINALLY, THE DATA ARE SCALED AND PLOTTED.
C
        DO 110 I=1,N1
          RI = I
          X(I) = PTTOPT*RI
          I1 = I+N0-1
110     Y(I) = KCAL*Y(I1)
        CALL SCALE(Y,4.,N1,+1)
        CALL SCALE(X,LENGTH,N1,+1)
        CALL NEWPEN(-1)
        CALL LINE(X,Y,N1,1,0,31)
        CALL NEWPEN(1)
        FIRST=Y(N1+1)-2.0*Y(N1+2)
        CALL AXIS(0.,-2.,'DISTANCE (MICROMETERS)',-22,LENGTH,0.,
```

```
          1  X(N1+1),X(N1+2))
             CALL AXIS(0.,-2.,'VERTICAL DISPLACEMENT (MICROMETERS)',
          1  35,8.,90.,FIRST,Y(N1+2))
             CALL SYMBOL(1.,7.5,.2,SPEC,0.,+80)
             CALL SYMBOL(1.,7.,.2,'PROF NO = ',0.,+10)
             CALL NUMBER(3.,7.,.2,RNUMBR,0.,-1)
             RN0 = N0
             RN2 = N0+N1-1
             CALL SYMBOL (4.,7.,.2,'PTS      TO',0.,+11)
             CALL NUMBER(4.8,7.,.2,RN0,0.,-1)
             CALL NUMBER(6.5,7.,.2,RN2,0.,-1)
             CALL PLOT(0.,0.,+23)
  300     CONTINUE
          CALL PLOT(0.,0.,999)
          STOP
          END
```

8.4 Flowchart for PLOTSVIL



The operator is prompted to type in the name of the data file to be examined.

The file is opened as a logical unit, and from it the program reads the specimen information, the point-to-point spacing, the calibration constant, and the total number of profiles.

The operator is prompted to type the total number of profiles to be plotted, the position of each one in the file and, for all the plots, the beginning data point, the number of data points, and the length of the plot output.

The Main Loop begins. Each profile is read.

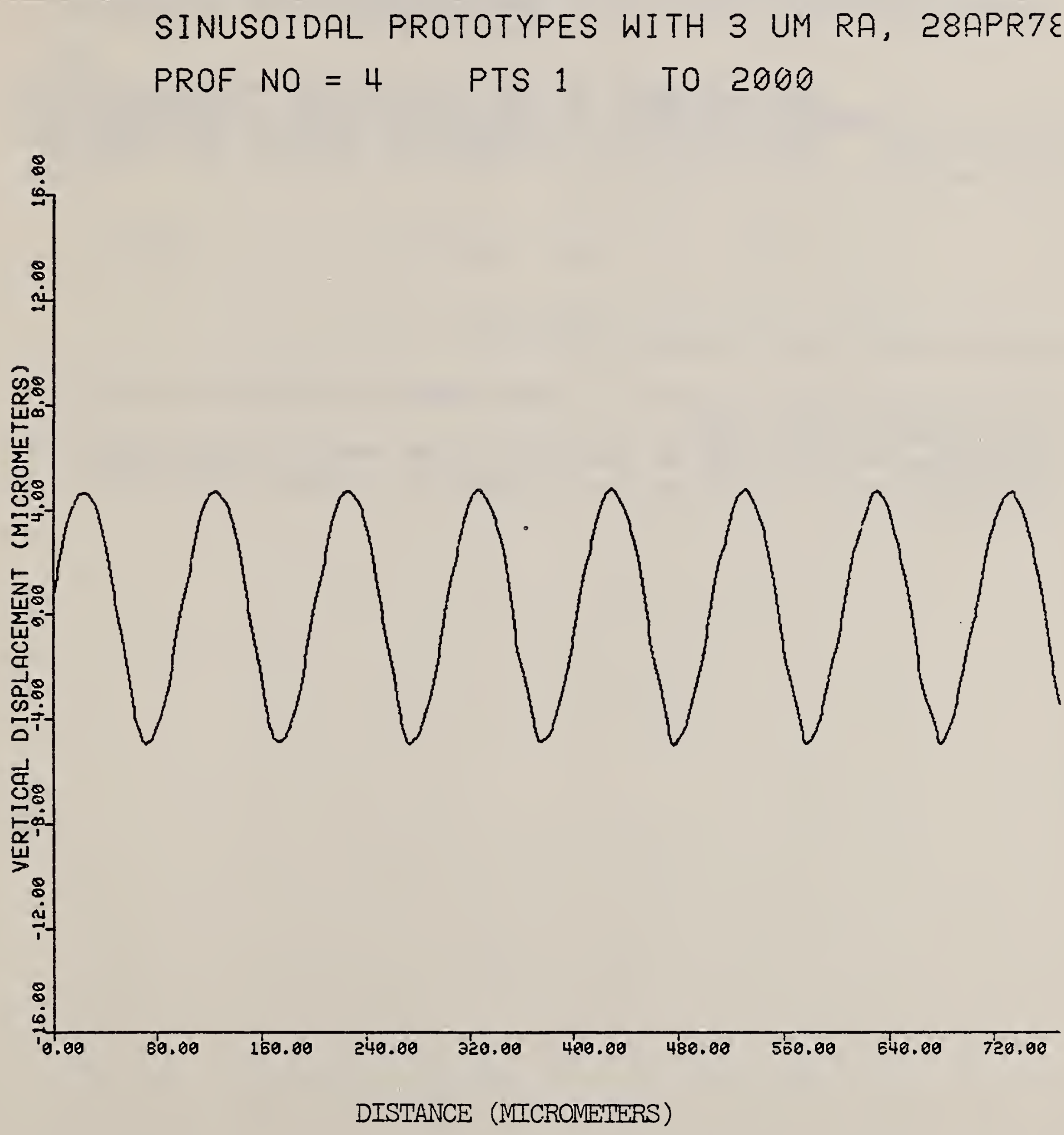
The average height of the profile is calculated as information for the operator.

The data points are scaled and plotted.

Close plot sequence and stop.

8.5 Example of PLOTSVIL Plot

The sinusoidal surface depicted here was also discussed in sec. 7.5. The ADF,ACF, and PSD statistical functions for this profile are given in secs. 9.5,10.5, and 11.5. Shown here is the first page of a graph which spans several pages.



9. ADF

9.1 Summary

The ADF calculates and plots a histogram of the surface heights y_i for a profile. The 4000 data points, whose quantization levels range from -2048 to 2047, are sorted into 256 boxes containing 16 quantization levels each. The formula is

$$ADF(j) = N_j/(16x4000xKCAL), \qquad (9.1)$$

where N_j is the number of data points with height values falling in the jth box. The height of the jth box y(j) is the abscissa of this function. It is given by

$$y(j) = 16\ KCAL\ (j-257/2). \qquad (9.2)$$

The operator has the following options:

1. choosing whether to fit a least squares straight line or a mean line to the data,
2. printing out the set of numbers which comprise the function,
3. choosing which profiles to analyze in a given data file.

9.2 Operating System Commands

```
***** ADF *****
*
VPHS1 SRF:ADF,@1
VPHS2 SRF:ADF,3
$W*
$W*
$W*
$W* THE PLOTTING PROGRAM HAS BEEN COMPLETED.
$W* THE RASTER FILE "S:RASDATA.RAS" HAS BEEN CREATED
$W* AND MAY BE PLOTTED ON THE 3230 SYSTEM USING THE COMMAND "PLOTFINI".
$W*
$W* TO LOOK AT THE ENTIRE SET OF COMMANDS FOR
$W* MEASURING AND ANALYSING SURFACE PROFILE DATA,
$W* USE THE COMMAND "FASTMENU".
$EXIT
```

9.3 ADF FORTRAN Program

```
C ***** SRF:ADF.FTN CALLED BY "ADF" *****
C THIS ROUTINE CALCULATES THE ADF FOR A SET OF N=4000 PROFILE
C DATA POINTS RANGING FROM -2048 TO +2047. THESE ARE  SORTED
C INTO K=256 BOXES. THE NUMBER OF POINTS IN EACH BOX  IS
C ACUMULATED INTO THE ARRAY  ADF. THE ARRAY ORDNIT GIVES
C THE MIDPOOINT IN EACH BOX. THE PROCEDURE IS REPEATED FOR
C EACH SET OF PROFILE DATA. AN AVERAGE ADF, AVADF, IS ALSO
C CALCULATED AND PLOTTED.
C         T. VORBURGER (5/24/78)   REVISED (3/82)
C
C
C
      DIMENSION ORDNIT(258),ADF(258),AVADF(258)
      DIMENSION SPEC(15)
      INTEGER START
      REAL KCAL
      INTEGER*2 PROFIL(4000),DATFIL(9)
      INTEGER*2 QUERY1,QUERY2,QUERY3,PRONUM(50)
      DATA DATFIL(7),DATFIL(8),DATFIL(9)/'.D','AT','  '/
C
1     FORMAT(//' SHOULD WE DO A LEAST SQUARES STRAIGHT LINE'/
     1' FIT TO THE DATA? "Y" OR "N"?')
2     FORMAT(40A2)
3     FORMAT(20A4)
4     FORMAT(E13.6)
5     FORMAT(E13.6,' UM/QUANTIZATION LEVEL = KCAL')
6     FORMAT(1H ,20A4)
7     FORMAT('*** ADF CALCULATIONS ***'/
     1' WHAT IS THE FILE NAME?'/' THE FORMAT SHOULD',
     2' BE "VOL:NAME" WITH EXACTLY 12 CHARACTERS.')
8     FORMAT(' THE FIRST & LAST PROFILE POINTS ARE ',I5,' & ',I5)
11    FORMAT(' POINT-TO-POINT SPACING (16 KCAL) = ',E13.6,' UM')
12    FORMAT(' TOTAL PROBABILITY = ',F8.4,' UM')
13    FORMAT(' RA CALCULATED FROM ADF = ',F9.5,' UM')
14    FORMAT(' RMS ROUGHNESS CALCULATED FROM ADF = ',
     1F9.5,' UM')
16    FORMAT(8E12.4)
17    FORMAT(16I5)
18    FORMAT(I2)
19    FORMAT(' I CAN"T READ THIS NAME. TRY AGAIN.')
21    FORMAT(1H /'ADF DATA FOR POSITION ',I2)
22    FORMAT(1H /'AVERAGE ADF FOR ALL ',I2,' POSITIONS')
23    FORMAT(' ERROR ON OPENW ROUTINE'/
     1' TYPE THE FILE NAME OVER AND GET IT RIGHT THIS TIME.')
24    FORMAT(//' THE DATA ARE BEING FITTED TO A'/
     1' LEAST SQUARES STRAIGHT LINE FOR THIS'/
     2' CALCULATION.')
25    FORMAT(//' A MEAN VALUE IS BEING SUBTRACTED',
     1' FROM THE DATA.')
27    FORMAT(1H ,40A2)
28    FORMAT(1H1,5X,' ***** ADF CALCULATIONS *****'///)
29    FORMAT(//' DO YOU WANT A PRINTOUT OF THE '/
     1' ADF NUMBERS? "Y" OR "N"?')
31    FORMAT(1H1,'***** WARNING! *****'/' SOME OF THE'
     1' PROFILE POINTS ARE OUTSIDE THE USUAL RANGE'/
     2' OF -2047 TO 2048. YOU WILL GET SOME ERRORS'/
     3' FOR THE ADF OF PROFILE', I3,'.')
C
```

```
C
C FIRST, THE PROGRAM READS THE NAME OF THE DATA FILE TO BE
C ANALYZED FROM LU-5. THE NAME OF THE DATA FILE MUST HAVE
C THE FORMAT "VOL:ADATE", WHERE "VOL" IS THE 3 CHARACTER
C VOLUME NAME AND "ADATE" IS THE 8 CHARACTER FILE NAME.
C THE DATA FILE IS ASSIGNED LU-10. THEN THE PROGRAM
C READS THE SPECIMEN INFO, THE VERTICAL KCAL, AND THE
C PROFILE DATA FROM THE FILE "VOL:ADATE.DAT".
C
C
      ITRY = 0
92    WRITE(6,7)
      READ(5,2,ERR=93,END=502)(DATFIL(J),J=1,6)
      CALL CARCON(3,1)
      GO TO 94
93    ITRY = ITRY + 1
      IF (ITRY .GT. 2) GO TO 502
      WRITE(6,19)
      GO TO 92
94    CALL OPENW(10,DATFIL,0,0,0,ISTAT)
      IF(ISTAT.LT.1)GO TO 90
      WRITE(6,23)
      ITRY = ITRY + 1
      IF (ITRY .GT. 2) GO TO 502
      GO TO 92
C
C
90    READ(10,3,REC=2)SPEC
      READ(10,4,REC=6)KCAL
      K = 256
      KPLUS = K+2
      N = 4000
      CONST = 16.*KCAL
      IAVRG = 0
      WRITE (6,1)
      READ (5,2) QUERY1
      WRITE(3,28)
      WRITE(3,27) DATFIL
      WRITE(3,6) SPEC
      WRITE(3,5) KCAL
      IF (QUERY1 .EQ. 'Y ') WRITE (3,24)
      IF (QUERY1 .EQ. 'N ') WRITE (3,25)
      WRITE(6,29)
      READ(5,2) QUERY3
C
        DO 110 J=1,K
        ORDNIT(J) = CONST*(J-(K+1)/2.)
110     AVADF(J) =0.
C
      CALL PLOTS(0,0,0)
      CALL FACTOR(.7874)
      CALL PLOT(.5,.5,-3)
      READ(10,18,REC=7) NTOT
C
C
C NOW, THE OPERATOR IS ASKED TO CHOOSE WHETHER ALL
C THE PROFILES ARE TO BE ANALYZED AND, IF NOT, HOW MANY.
C
```

```
C
      CALL CHOICE(NTOT,ITOPRO,PRONUM)
C
C
C THE MAIN LOOP TO CALCULATE AND PLOT THE ADF FOR EACH
C PROFILE BEGINS HERE
C
C
      DO 300 M = 1,ITOPRO
      RNUMBR = PRONUM(M)
      START = (PRONUM(M) - 1)*251 + 8
      READ (10,18,REC=START) NRUN
      READ (10,17) PROFIL
      WRITE(6,8) PROFIL(1),PROFIL(4000)
C
      IF (QUERY1 .EQ. 'Y ') CALL LEASQ (PROFIL,4000)
      IF (QUERY1 .EQ. 'N ') CALL MEAN (PROFIL,4000)
C
C
C NOW, WE CALCULATE THE NORMALIZED ADF AND THE CORRESPOND-
C ING ARRAY OF ORDINATE VALUES (IN UM). THE RESULTS ARE
C CHECKED BY CALCULATING THE RA AND THE RMS ROUGHNESS
C FROM THE ADF RESULTS.
C
C
        MESSG = 0
        DO 101 J=1,K
101     ADF(J) = 0.
C
        DO 102 I= 1,N
        BOX = .5*K*(1.+(PROFIL(I)/2048.))
        IBOX = BOX+1.
        IF (IBOX .LE. K .AND. IBOX .GE. 1) GO TO 102
        IF (MESSG .GT. 0) GO TO 105
        MESSG = 1
        WRITE(3,31) NRUN
105     IF(IBOX .GT. K) IBOX=K
        IF(IBOX .LT. 1) IBOX=1
102     ADF(IBOX) =   ADF(IBOX) + 1.
C
        DO 103 J=1,K
        ADF(J) = ADF(J)/(CONST*4000.)
103     AVADF(J) = AVADF(J) + ADF(J)/ITOPRO
C
        SUM0 = 0.
        SUM1 = 0.
        SUM2 = 0.
        DO 104 J=1,K
        SUM0 = SUM0 + ADF(J)
        SUM1 = SUM1 +ADF(J)*ABS(ORDNIT(J))
104     SUM2 = SUM2+ADF(J)*(ORDNIT(J)**2)
C
      UNITY = CONST*SUM0
      RACHK = CONST*SUM1
      RMSCHK = SQRT(CONST*SUM2)
C
C
C NOW, WE WRITE THE RESULTS ON THE PRINTER
```

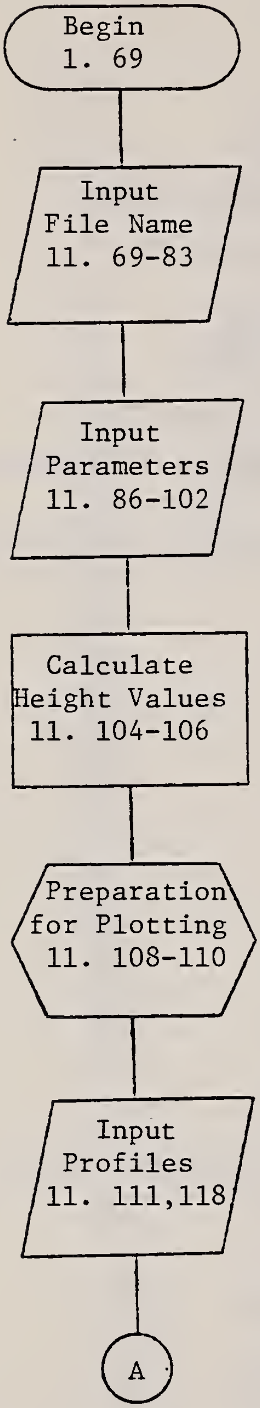
```
C IF CALLED FOR.
C
C
C
      IF (QUERY3 .EQ. 'N ') GO TO 601
      CALL CARCON(3,0)
      CALL CARCON(3,1)
      WRITE(3,21) NRUN
      WRITE(3,12)UNITY
      WRITE(3,13)RACHK
      WRITE(3,14)RMSCHK
      WRITE(3,11)CONST
      WRITE(3,16)(ADF(I),I=1,256)
C
C
C NOW, WE PLOT THE ADF.
C
C
601   CALL PLOTME(ORDNIT,ADF,KPLUS,'ADF (INVERSE UM)',16,
     1'HEIGHT FROM MEAN LINE (UM)',26,SPEC,RMSCHK,
     1NRUN,IAVRG)
300   CONTINUE
      CALL CLOSE (10,STATUS)
C
C
C THE MAIN LOOP IS COMPLETED. NOW, WE PRINT AND PLOT THE
C RESULTS FOR THE AVERAGE OF ALL RUNS.
C
C
        IF(ITOPRO .EQ. 1) GO TO 501
        IAVRG = 1
        SUMAV0 = 0.
        SUMAV1 = 0.
        SUMAV2 = 0.
        DO 301 J=1,K
        SUMAV0 = SUMAV0+AVADF(J)
        SUMAV1 = SUMAV1+AVADF(J)*ABS(ORDNIT(J))
301     SUMAV2 = SUMAV2+AVADF(J)*(ORDNIT(J)**2)
C
      AVONE = CONST*SUMAV0
      AVRA =  CONST*SUMAV1
      AVRMS = SQRT(CONST*SUMAV2)
      CALL CARCON(3,0)
      CALL CARCON(3,1)
      WRITE(3,22) ITOPRO
      WRITE(3,12) AVONE
      WRITE(3,13) AVRA
      WRITE(3,14) AVRMS
      WRITE(3,11) CONST
      WRITE(3,16) (AVADF(J),J=1,256)
      CALL PLOTME(ORDNIT,AVADF,KPLUS,'ADF (INVERSE UM)',16,
     1'HEIGHT FROM MEAN LINE (UM)',26,SPEC,AVRMS,ITOPRO,IAVRG)
501   CALL PLOT(0.,0.,999)
502   STOP
      END
C
C
C SUBROUTINE FOR PLOTTING
```

```
C
C
      SUBROUTINE PLOTME(X,Y,KPLUS,YLABEL,NYCHAR,XLABEL,
     1NXCHAR,SPEC,RMS,NRUN,IAVRG)
C
C THE INPUT VARIABLES IN THIS SUBROUTINE ARE:
C X - THE ARRAY OF X VALUES,
C Y - THE ARRAY OF Y VALUES,
C KPLUS - THE SIZE OF THE ARRAYS PLUS 2 FOR STORAGE OF
C         THE SCALE FACTOR AND INTIAL POINT IN THE PLOT,
C YLABEL, XLABEL - THE LABELS FOR THE X AND Y AXES,
C NXCHAR,NYCHAR - THE NUMBER OF CHARACTERS IN EACH LABEL,
C SPEC - A TITLE LABEL FOR THE PLOT,
C RMS - THE VALUE FOR THE RMS ROUGHNESS CALCULATED FROM
C       THE FUNCTION,
C NRUN - THE PROFILE'S NUMBER IN THE INPUT DATA FILE,
C IAVRG - A FLAG TO TELL THE PROGRAM WHETHER AN
C         INDIVIDUAL FUNCTION OR THE AVERAGE IS BEING
C         CALCULATED.
C
      DIMENSION X(KPLUS),Y(KPLUS),XLABEL(20),YLABEL(20)
      DIMENSION SPEC(15)
      RUN = NRUN
      K = KPLUS-2
      CALL SCALE(Y,8.,K,+1)
      CALL SCALE(X,12.,K,+1)
      CALL NEWPEN(-1)
      Y(K+2) = -Y(K+2)
      Y(K+1) = Y(K+1) - 8.*Y(K+2)
      CALL LINE(Y,X,K,1,0,31)
      CALL AXIS(0.,0.,YLABEL,-NYCHAR,9.,0.,Y(K+1),Y(K+2))
      CALL AXIS(9.,0.,XLABEL,-NXCHAR,12.,90.,X(K+1),X(K+2))
      CALL SYMBOL(0.,0.2,.2,SPEC,90.,60)
      CALL SYMBOL(0.5,.2,.2,'RMS ROUGHNESS = ',90.,16)
      IF ( RMS .LT. 0.01) GO TO 501
      CALL NUMBER(0.5,3.4,.2,RMS,90.,+3)
      CALL SYMBOL (0.4,4.6,0.2,7,90.,-1)
      CALL SYMBOL (0.4,4.8,0.2,109,90.,-1)
      GO TO 502
501   CALL NUMBER (0.5,3.4,0.2,RMS,90.,+5)
      CALL SYMBOL (0.4,5.0,0.2,7,90.,-1)
      CALL SYMBOL(0.4,5.2,0.2,109,90.,-1)
502   IF (IAVRG.EQ.1) GO TO 400
      CALL SYMBOL(1.,.2,.2,'ADF FOR POSITION',90.,16)
      CALL NUMBER(1.,3.7,.2,RUN,90.,-1)
      GO TO 401
400   CALL SYMBOL(1.,.2,.2,'AVERAGE ADF FOR ALL ',90.,20)
      CALL NUMBER(1.,4.3,.2,RUN,90.,-1)
      CALL SYMBOL(1.,4.7,.2,' POSITIONS',90.,10)
401   CALL PLOT(0.,0.,+23)
      RETURN
      END
C
C
C SUBROUTINE FOR DETERMINING WHICH PROFILES IN
C THE FILE ARE ANALYZED
C
C
```

```
      SUBROUTINE CHOICE(NTOT,ITOPRO,PRONUM)
      INTEGER*2 QUERY2,PRONUM(50)
C
C
C INPUT VARIABLES:
C NTOT - THE TOTAL NUMBER OF PROFILES IN THE FILE.
C OUTPUT VARIABLES:
C ITOPRO - THE NUMBER OF PROFILES THAT THE OPERATOR CHOOSES
C          TO ANALYZE.
C PRONUM - THE ARRAY WHICH CONTAINS THE POSITIONS OF THE
C          CHOSEN PROFILES IN THE DATA FILE.
C
1     FORMAT(' THERE ARE',I3,' PROFILES'//' DO YOU',
     1' WANT TO ANALYZE ALL OF THEM?'/' "Y" OR "N"?')
2     FORMAT(I2)
3     FORMAT(//' HOW MANY PROFILES DO YOU WANT TO ANALYZE?'/
     1' USE I2 FORMAT.')
4     FORMAT(//' TYPE THE NUMBER OF THE FIRST PROFILE TO BE ANALYZED.'/
     1' USE I2 FORMAT.')
5     FORMAT(A2)
6     FORMAT(//' TYPE THE NUMBER OF THE NEXT PROFILE TO BE ANALYZED.'/
     1' USE I2 FORMAT.')
C
      WRITE (6,1) NTOT
      READ (5,5) QUERY2
      IF ( QUERY2 .EQ. 'N ') GO TO 131
C
      ITOPRO = NTOT
      DO 132 I = 1,ITOPRO
132   PRONUM(I) = I
      RETURN
C
131   WRITE (6,3)
      READ (5,2) ITOPRO
      WRITE (6,4)
      READ (5,2) PRONUM(1)
      DO 133 I=2,ITOPRO
      WRITE (6,6)
133   READ (5,2) PRONUM(I)
      RETURN
      END
```

The subroutines, MEAN and LEASQ, have already been listed in the program SMORGAS.

9.4 Flowchart for ADF



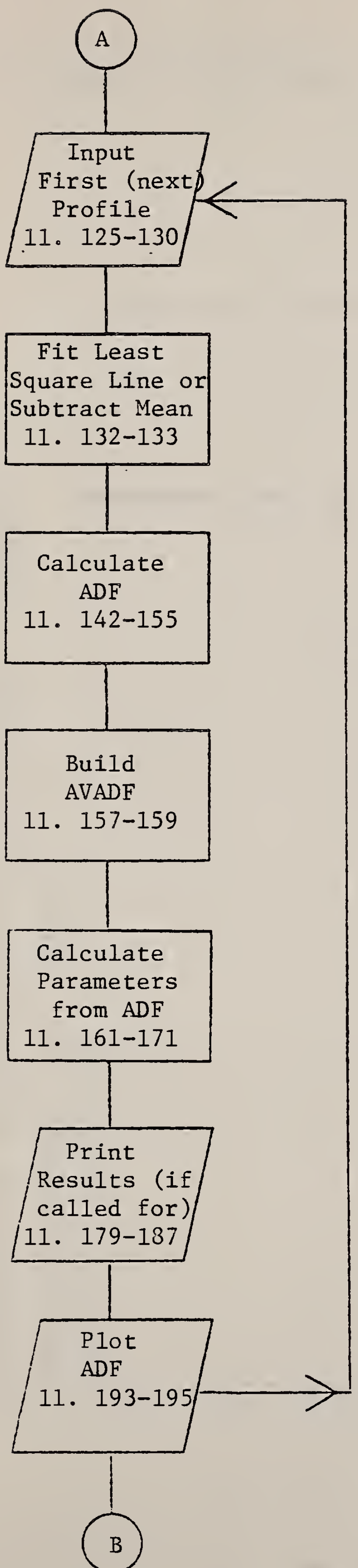
The operator is prompted to type the name of the data file to be examined. The file is then opened as a logical unit.

The program reads the data file for the specimen I.D. and the instrument calibration constant and prompts the operator to choose whether or not to use a least squares fitted line as the mean line. If not the mean line is simply the mean of the data values. The operator much also choose whether or not to print out the set of numbers which comprise the ADF.

The set of height values, the independent variable in the ADF, are calculated here.

Three subroutines are called to initialize the plotting sequence.

The total number of profiles in the file is read. Then the operator is asked to choose which of these will be analyzed. The subroutine CHOICE is called for this task.



The main loop to calculate and plot the ADF begins.

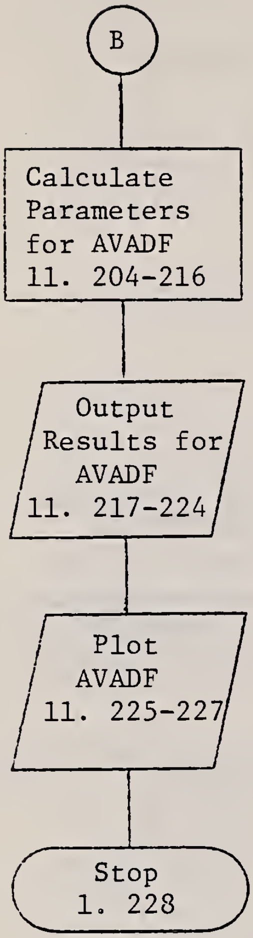
A subroutine, either LEASQ or MEAN, is called.

This section includes a set of statements (ll. 150-154) to warn the operator and massage the data when data points are out of range. This can happen when using the least squares fit with large amplitude data.

An average ADF is being calculated by adding each function to a summing array.

Three quantities are calculated to check the accuracy of the ADF: R_a, R_q, and the sum of the ADF values, which should equal unity.

The subroutine PLOTME is called here.

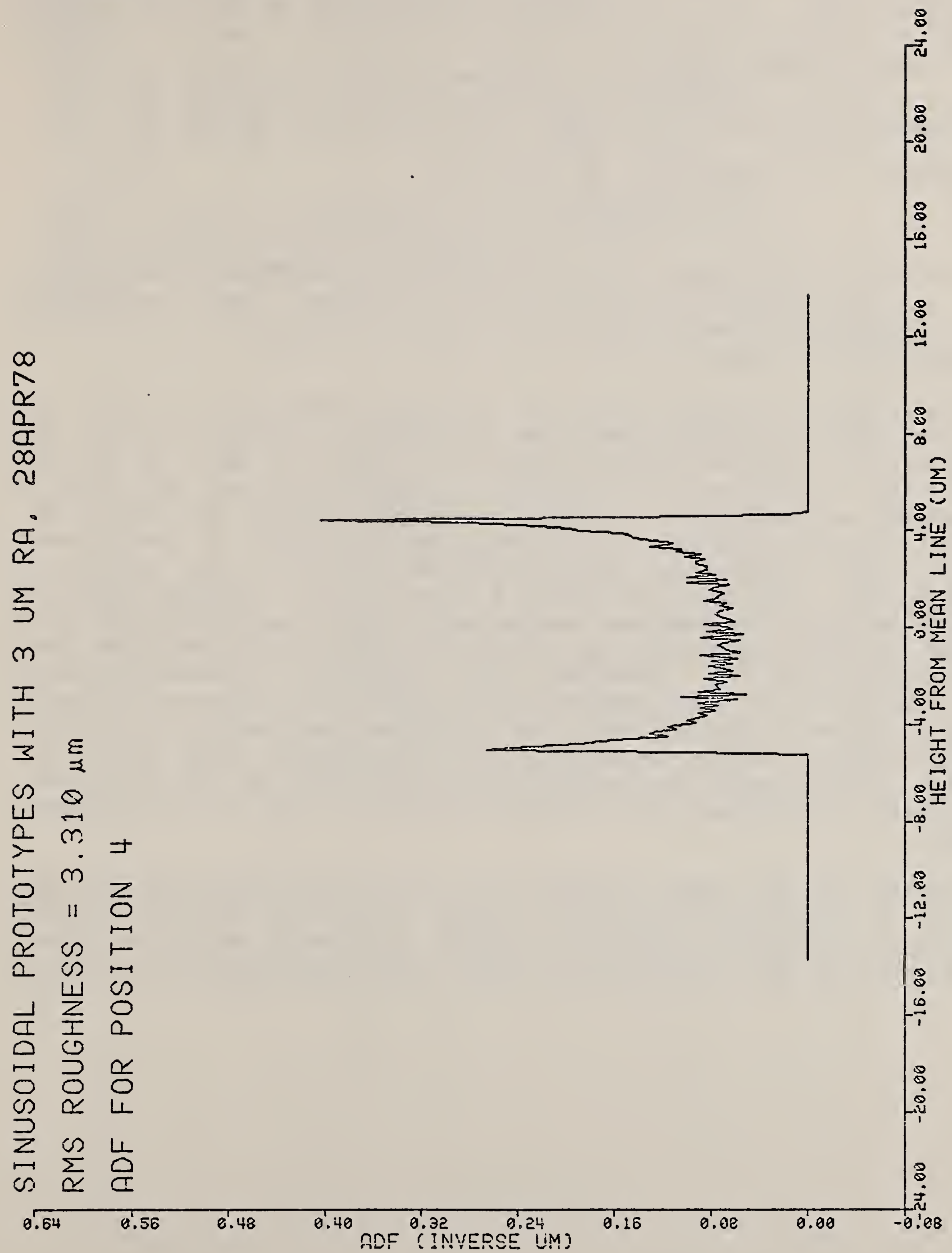


R_a, R_q, and the sum of the ADF values are calculated for AVADF.

The AVADF function and the three parameters are now printed.

PLOTME is called again.

9.5 Example of ADF Plot for Sinusoidal Surface



10. ACF

10.1 Summary

The ACF calculates and plots the autocorrelation function from the profile data. It is a quantitative measure of the similarity between a laterally shifted and an unshifted version of the profile and is given by

$$ACF(kT) = 1/[(N-k)R_q^2] \sum_{i=1}^{N-k} y_i y_{i+k}, \qquad (10.1)$$

where T is the point-to-point spacing of the profile, k equals 0, 2, 4 . . . , and hence kT is the shift distance. The factor R_q^2 in the denominator normalizes the value of the ACF to unity at a shift distance of zero.

The above formula is known as the unbiased estimator [12] of the autocorrelation function because it divides the sum by N-k to compensate for the decreasing number of terms in the sum as the shift index k increases. This estimator is not as widely recommended as the biased estimator which divides the sum by N, because it can be shown that the mean square error of the biased estimator is generally smaller than that of the unbiased estimator [12]. These errors should be small for random surfaces typically studied in surface metrology where the characteristic spacings of surface features should be much smaller than the length of the profile. Moreover, for calibration specimens with highly periodic profiles, such as the sinusoidal specimen already discussed in secs. 8.5 and 9.5, the unbiased estimator (eq. 10.1) is significantly better over the shift distance range shown in fig. 10.5. The biased estimator incorrectly yields an ACF whose oscillations would damp out significantly as the shift distance increases.

The operator has the same options here as in the ADF program. The function is calculated only for shift distances that are even multiples of the point spacing in order to save calculation time. With 501 points the calculation itself takes approximately 200 s.

10.2 Operating System Commands

```
***** ACF *****
*
VPHS1 SRF:ACF,@1
VPHS2 SRF:ACF,3
$W*
$W*
$W*
$W* THE PLOTTING PROGRAM HAS BEEN COMPLETED.
$W* THE RASTER FILE "S:RASDATA.RAS" HAS BEEN CREATED
$W* AND MAY BE PLOTTED ON THE 3230 SYSTEM USING THE COMMAND "PLOTFINI".
$W*
$W* TO LOOK AT THE ENTIRE SET OF COMMANDS FOR
$W* MEASURING AND ANALYSING SURFACE PROFILE DATA,
$W* USE THE COMMAND "FASTMENU".
$EXIT
```

10.3 ACF FORTRAN Program

```
C ***** SRF:ACF.FTN CALLED BY "ACF" *****
C THIS ROUTINE CALCULATES THE ACF FOR A SET OF N=4000 PROFILE
C DATA POINTS . THE PROCEDURE IS REPEATED FOR EACH SET OF PROFILE
C DATA CHOSEN TO BE ANALYZED. AN AVERAGE ACF IS ALSO CALCULATED
C AND PLOTTED.
C           T. VORBURGER (5/24/78)   REVISED (7/82)
C
C
C
        DIMENSION ACF(503),AVACF(503)
        DIMENSION SPEC(15),X(503)
        INTEGER START
        REAL KCAL
        INTEGER*2 PROFIL(4000),DATFIL(9),N,ISHIFT,JPLUS
        INTEGER*2 QUERY1,QUERY2,QUERY3,PRONUM(50)
        DATA DATFIL(7),DATFIL(8),DATFIL(9)/'.D','AT','  '/
C
1       FORMAT(///' SHOULD WE DO A LEAST SQUARES STRAIGHT LINE'/
       1' FIT TO THE DATA? "Y" OR "N"?')
2       FORMAT(40A2)
3       FORMAT(20A4)
4       FORMAT(E13.6)
5       FORMAT(E13.6,' UM/QUANTIZATION LEVEL = KCAL')
6       FORMAT(1H ,20A4)
7       FORMAT(5X,'*** ACF CALCULATIONS ***'/
       1' WHAT IS THE FILE NAME?'/' THE FORMAT SHOULD',
       2' BE "VOL:NAME" WITH EXACTLY 12 CHARACTERS.')
8       FORMAT(I2,F10.4)
9       FORMAT(' APPROX RMS SLOPE (FROM ACF) =',F6.4,'-',F6.4)
10      FORMAT(F10.4)
11      FORMAT(' POINT-TO-POINT SPACING =',F9.4,' UM')
12      FORMAT(' POINT-TO-POINT SPACING OF ACF =',F10.4,' UM')
13      FORMAT(' RA CALCULATED FROM ORIGINAL DATA = ',F9.5,' UM')
14      FORMAT(' RMS ROUGHNESS CALCULATED FROM ACF = ',
       1F9.5,' UM')
15      FORMAT(I3)
17      FORMAT(16I5)
18      FORMAT(I2)
19      FORMAT(' I CAN"T READ THIS NAME. TRY AGAIN.')
20      FORMAT(' START LOOP')
21      FORMAT(1H /' ACF DATA FOR POSITION ',I2)
22      FORMAT(1H /' AVERAGE ACF FOR ALL ',I2,' POSITIONS')
23      FORMAT(' ERROR ON OPENW ROUTINE'/
       1' TYPE THE FILE NAME OVER AND GET IT RIGHT THIS TIME.')
24      FORMAT(///' THE DATA ARE BEING FITTED TO A'/
       1' LEAST SQUARES STRAIGHT LINE FOR THIS'/
       2' CALCULATION.')
25      FORMAT(///' A MEAN VALUE IS BEING SUBTRACTED',
       1' FROM THE DATA.')
26      FORMAT(' COMPLETE LOOP')
27      FORMAT(1H ,40A2)
28      FORMAT(1H1,5X,' ***** ACF CALCULATIONS *****'///)
29      FORMAT(///' DO YOU WANT A PRINTOUT OF THE '/
       1' ACF NUMBERS? "Y" OR "N"?')
30      FORMAT(' HOW MANY LAG POINTS DO YOU WANT TO CALCULATE?'/
       1' THE UPPER LIMIT IS 501.')
31      FORMAT(' THE FIRST AND LAST PROFILE POINTS ARE',I5,' & ',I5)
32      FORMAT(///' THE RMS SLOPE CANNOT BE CALCULATED FROM'/
```

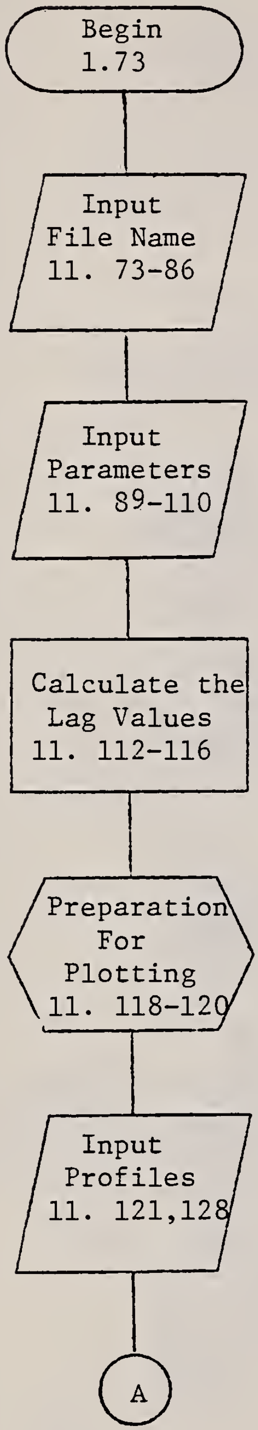
```
     1' THE ACF FOR THIS PROFILE BECAUSE SMALL ERRORS'/
     2' IN THE CALCULATION LEAD TO A NEGATIVE SQUARE ROOT.')
33    FORMAT(10F8.4)
C
C
C FIRST, THE PROGRAM READS THE NAME OF THE DATA FILE TO BE
C ANALYZED FROM LU-5. THE NAME OF THE DATA FILE MUST HAVE
C THE FORMAT "VOL:ADATE", WHERE "VOL" IS THE 3 CHARACTER
C VOLUME NAME AND "ADATE" IS THE 8 CHARACTER FILE NAME.
C THE DATA FILE IS ASSIGNED LU-10. THEN THE PROGRAM
C READS THE SPECIMEN INFO, THE VERTICAL KCAL, AND THE
C PROFILE DATA FROM THE FILE "VOL:ADATE.DAT".
C
C
      ITRY = 0
92    WRITE(6,7)
      READ(5,2,ERR=93,END=502)(DATFIL(J),J=1,6)
      GO TO 94
93    ITRY = ITRY + 1
      IF (ITRY .GT. 2) GO TO 502
      WRITE(6,19)
      GO TO 92
94    CALL OPENW(10,DATFIL,0,0,0,ISTAT)
      IF(ISTAT.LT.1)GO TO 90
      WRITE(6,23)
      ITRY = ITRY + 1
      IF (ITRY .GT. 2) GO TO 502
      GO TO 92
C
C
90    WRITE (6,30)
      READ (5,15) K
      READ(10,3,REC=2)SPEC
      READ(10,10,REC=4) PTTOPT
      READ(10,4,REC=6)KCAL
      KPLUS = K+2
      N = 4000
      AVACF0=0.
      IAVRG = 0
      WRITE (6,1)
      READ (5,2) QUERY1
      WRITE(3,28)
      WRITE(3,27) DATFIL
      WRITE(3,6) SPEC
      WRITE (6,11) PTTOPT
      DUBLPT = 2.*PTTOPT
      WRITE(3,12) DUBLPT
      WRITE(3,5) KCAL
      IF (QUERY1 .EQ. 'Y ') WRITE (3,24)
      IF (QUERY1 .EQ. 'N ') WRITE (3,25)
      WRITE(6,29)
      READ(5,2) QUERY3
C
        DO 113 I=1,K
        ISHIFT = 2*I - 2
        X(I) =  PTTOPT*ISHIFT
        AVACF(I) = 0.
113     CONTINUE
```

```
C
      CALL PLOTS(0,0,0)
      CALL FACTOR(.7874)
      CALL PLOT(.5,.5,-3)
      READ(10,18,REC=7) NTOT
C
C
C NOW, THE OPERATOR IS ASKED TO CHOOSE WHETHER ALL
C THE PROFILES ARE TO BE ANALYZED AND, IF NOT, HOW MANY.
C
C
      CALL CHOICE(NTOT,ITOPRO,PRONUM)
C
C
C THE MAIN LOOP TO CALCULATE AND PLOT THE ACF FOR EACH
C PROFILE BEGINS HERE
C
C
      DO 300 M = 1,ITOPRO
      RNUMBR = PRONUM(M)
      START = (PRONUM(M) - 1)*251 + 8
      READ (10,8,REC=START) NRUN,RA
      READ (10,17) PROFIL
      WRITE(6,31) PROFIL(1),PROFIL(4000)
C
      IF (QUERY1 .EQ. 'Y ') CALL LEASQ (PROFIL,4000)
      IF (QUERY1 .EQ. 'N ') CALL MEAN (PROFIL,4000)
C
C
C NOW WE DO THE DOUBLE LOOP IN THE AUTOCORRELATION CALCULATION.
C
C
      WRITE (6,20)
        DO 115 I=1,K
        ISHIFT = 2*I - 2
        SMPROD = 0.
        MTOP = N-ISHIFT
        RM= MTOP
          DO 112 J=1,MTOP
          JPLUS = J+ISHIFT
112       SMPROD = SMPROD+PROFIL(J)*PROFIL(JPLUS)
        SMTEMP = SMPROD/RM
        IF (I.GT.1) GO TO 111
        ACF0 = SMTEMP
        AVACF0 = AVACF0 + ACF0/ITOPRO
111     ACF(I) = SMTEMP/ACF0
115    AVACF(I) = AVACF(I) + ACF(I)/ITOPRO
      WRITE(6,26)
      RMS = SQRT(ACF0)*KCAL
      IF (ACF(2).GT.1 .OR. ACF(3).GT.1) GO TO 117
      SLOPE1 = RMS*SQRT(2.*(1.-ACF(2)))/DUBLPT
      SLOPE2 = 0.5*RMS*SQRT(2.*(1.-ACF(3)))/DUBLPT
C
C
C NOW, WE WRITE THE ACF VALUES ONTO THE PRINTER IF CALLED FOR.
C
C
117   WRITE(3,21) NRUN
```

```
      WRITE(3,13)RA
      WRITE(3,14)RMS
      IF (ACF(2).GT.1 .OR. ACF(3).GT.1) GO TO 116
      WRITE(3,9)SLOPE1,SLOPE2
      GO TO 118
116   WRITE(3,32)
118   IF (QUERY3 .EQ. 'N') GO TO 601
      WRITE(3,33)(ACF(I),I=1,K)
C
C
C NOW, WE PLOT THE ACF.
C
C
601   CALL PLOTME(X,ACF,KPLUS,'NORMALIZED ACF',14,
     1'SHIFT DISTANCE (UM)',19,SPEC,RMS,
     1NRUN,IAVRG)
300   CONTINUE
      CALL CLOSE (10,STATUS)
C
C
C THE MAIN LOOP IS COMPLETED. NOW, WE PRINT AND PLOT THE
C RESULTS FOR THE AVERAGE OF ALL RUNS.
C
C
      IF(ITOPRO .EQ. 1) GO TO 501
      IAVRG = 1.
      AVRMS = SQRT(AVACF0)*KCAL
      AVSLP1 = AVRMS*SQRT(2.*(1.-AVACF(2)))/DUBLPT
      AVSLP2 = 0.5*AVRMS*SQRT(2.*(1.-AVACF(3)))/DUBLPT
      CALL CLOSE(9,STATUS)
      WRITE(3,22) ITOPRO
      WRITE(3,14) AVRMS
      WRITE(3,9) AVSLP1,AVSLP2
      WRITE(3,33) (AVACF(I),I=1,K)
      CALL PLOTME(X,AVACF,KPLUS,'NORMALIZED ACF',14,
     1'SHIFT DISTANCE (UM)',19,SPEC,AVRMS,ITOPRO,IAVRG)
501   CALL PLOT(0.,0.,999)
502   STOP
      END
```

The subroutines, MEAN and LEASQ, have been listed already in the SMORGAS program. CHOICE and PLOTME have been listed in the ADF program. The PLOTME subroutine in ACF differs from the other one by a single row of text in an output label.

10.4 Flowchart for ACF



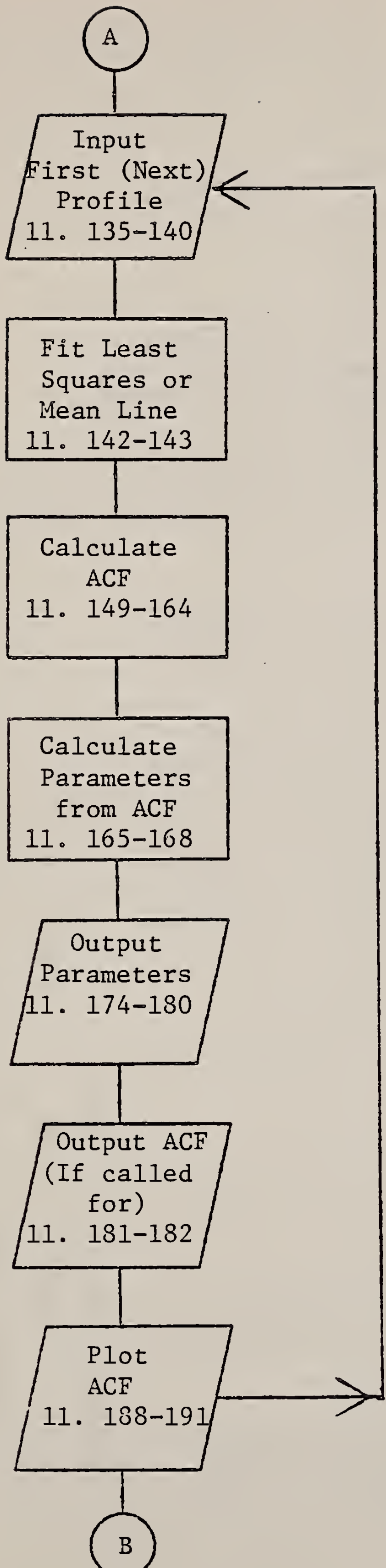
The operator is prompted to type the name of the data file to be examined. The file is then opened as a logical unit.

The operator is prompted to type the number of points in the ACF. This value must range from 1 to 501. Then the program reads the specimen I.D., the horizontal spacing of the data points, and the instrument calibration constant and prompts the operator to choose whether or not to fit a least squares line to the data and whether to print out the values of the ACF function.

The lag values comprise the independent variable of the ACF.

Three subroutines are called to initialize the plotting sequence.

The total number of profiles in the data file is read. Then the operator is asked to choose which of these will be analyzed. The subroutine CHOICE is used for this task.



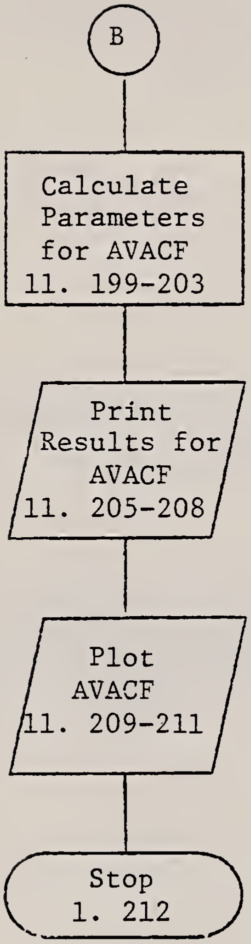
The main loop to calculate and plot the ACF begins. In addition to the profile data, the number of the profile and the stored value of R_a are also read from the file.

A subroutine, either LEASQ or MEAN, is called.

The ACF is calculated and normalized and the running average AVACF is built.

R_q and the rms slope are calculated from the ACF. In certain cases, a square root of a negative number may occur. If so, the calculation of rms slope is skipped.

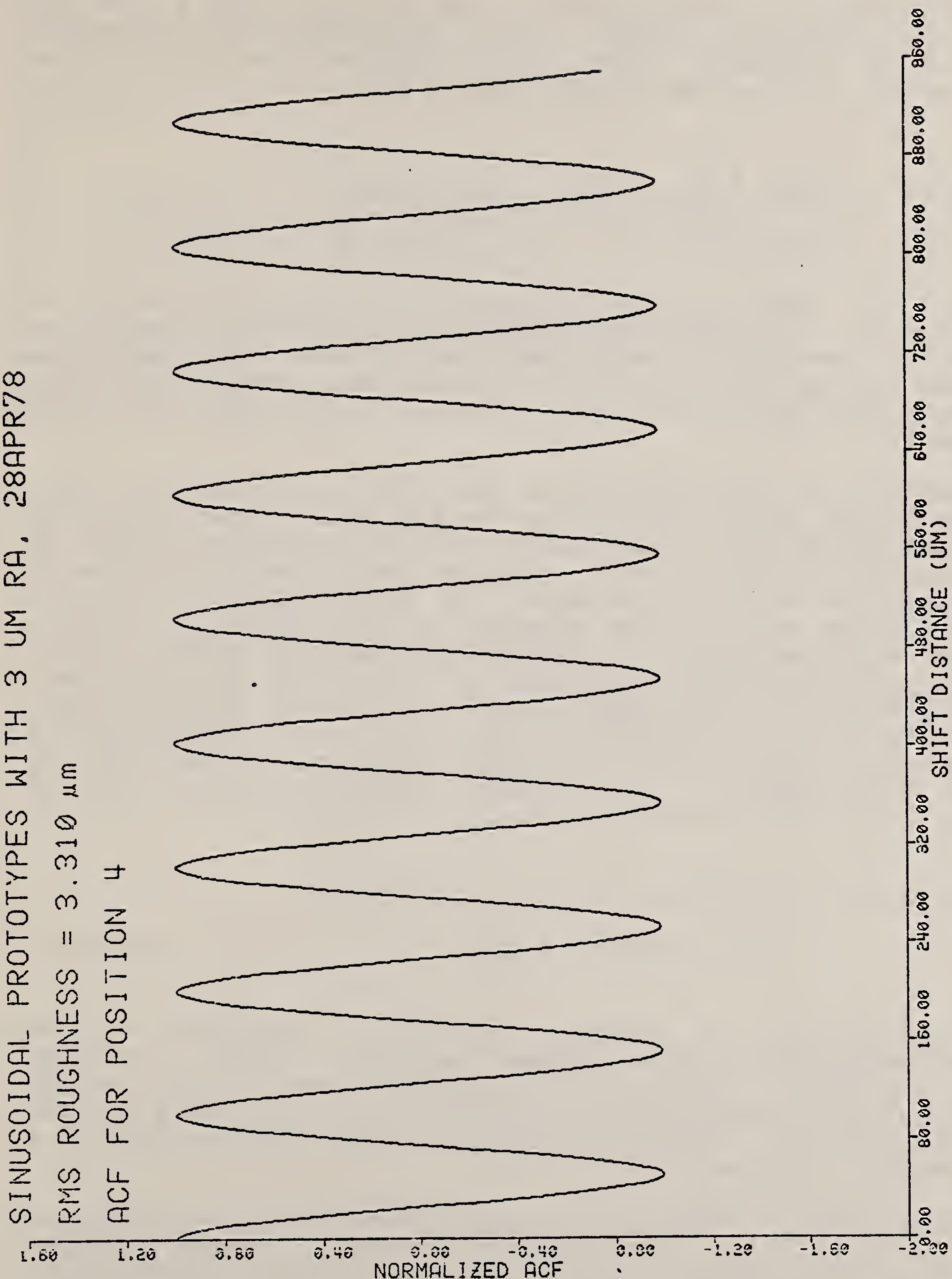
The subroutine PLOTME is called here.



R_q and rms slope are calculated from AVACF.

PLOTME is called again.

10.5 Example of ACF Plot



11. PSD

11.1 Summary

The power spectral density resolves the surface profile into its spatial frequency components. It is calculated by taking the square of the absolute value of the Fourier transform of the surface profile. The basic algorithm is given by

$$PSD(f) = (2T/N) \left| \sum_{i=1}^{N} y_i \exp[-j2\pi f(i-1)T] \right|^2 \qquad (11.1)$$

where $j = \sqrt{-1}$ and f is the spatial frequency with units of inverse length. The initial factor of 2 is included because we calculate a one-sided power spectrum ($0<f<\infty$) and ignore the components having $-\infty<f\leq 0$. However, it is important to note that there are other modifications of the basic algorithm due to our requirements.

The transform is calculated by a Fast Fourier Transform (FFT) algorithm called EZFFT supplied by the NBS GAMS library [13]. In the present application, the use of this FFT algorithm speeds up the calculation of the Fourier transform by at least a factor of 20 over the conventional approach. The EZFFT algorithm requires a large storage array which depends on the number of profile points to be transformed. In order to save user memory space, the 4000 point profile is divided into two 2000 point halves. The least squares fitted mean line is found for each section, then the PSD is calculated for each. The 2000 point profile yields a digitized PSD with 1000 digitized points, and this is smoothed to 500 points in the present calculation with a Hanning procedure [14]. The smoothing formula is given by

$$PSD(f=I\Delta) = 1/4\ PSD([I-1/2]\Delta) + 1/2\ PSD(I\Delta) + 1/4\ PSD([I+1/2]\Delta), \qquad (11.2)$$

where $\Delta(=2/NT)$ is the spacing in spatial frequency of the digitized points and I is an integer between 1 and 499. The last point in the PSD (I = 500) is left unsmoothed and is not plotted. The final PSD is calculated as the average of the PSD's for the two profile halves. The PSD calculation takes approximately 8 s on the present computer.

The operator has the option of choosing which profiles to analyze in a given data file. The PSD is plotted for each, and the operator has the option to print out the numbers for each PSD. The average PSD is automatically calculated, plotted, and printed as well.

11.2 Operating System Commands

```
***** PSD *****
*
SE SYS 14.5
VPHS1 SRF:PSD,@1
SE SYS 20
VPHS2 SRF:PSD,3
$W*
$W*
$W*
$W* THE PLOTTING PROGRAM HAS BEEN COMPLETED.
$W* THE RASTER FILE "S:RASDATA.RAS" HAS BEEN CREATED
$W* AND MAY BE PLOTTED ON THE 3230 SYSTEM USING THE COMMAND "PLOTFINI".
$W*
$W* TO LOOK AT THE ENTIRE SET OF COMMANDS FOR
$W* MEASURING AND ANALYSING SURFACE PROFILE DATA,
$W* USE THE COMMAND "FASTMENU".
$EXIT
```

11.3 PSD FORTRAN Program

```
C ***** SRF:PSD.FTN CALLED BY "PSD" *****
C THIS ROUTINE CALCULATES THE PSD FOR A SET OF 4000 PROFILE
C DATA POINTS. THIS IS DONE BY BREAKING THE PROFILE INTO
C 2000 POINT SECTIONS, FITTING A LEAST SQUARES STRAIGHT
C LINE TO EACH SECTION, AND CRANKING THROUGH AN FFT WHICH
C YIELDS A 1000 POINT TRANSFORM. THIS ARRAY IS SMOOTHED
C INTO A 500 POINT POWER SPECTRUM FOR EACH 2000 POINT
C PROFILE SECTION. THE TWO HALVES ARE THEN AVERAGED TO
C YIELD THE FINAL PSD.
C THE PROCEDURE IS REPEATED FOR EACH SET OF PROFILE
C DATA CHOSEN TO BE ANALYZED. AN AVERAGE PSD IS ALSO CALCULATED
C AND PLOTTED.
C THE FFT ROUTINES CAME VIA SALLY HOWE IN THE
C CENTER FOR APPLIED MATHEMATICS.
C         T. VORBURGER (1/8/83)
C
C
C
      DIMENSION PSD(500),PSD2(500),AVPSD(500),WSAVE(6015)
      DIMENSION SPEC(15),F(501),PSDLOG(501)
      INTEGER START
      REAL KCAL,LAMBDA
      INTEGER*2 PROFIL(2000),QUERY3,PRONUM(50)
      CHARACTER*16 DATFIL
      DATFIL(13:16) = '.DAT'
C
1     FORMAT(' THE DC LEVEL IS',E12.4,' um')
2     FORMAT(40A2)
3     FORMAT(20A4)
4     FORMAT(E13.6)
5     FORMAT(E13.6,' um/QUANTIZATION LEVEL = KCAL')
6     FORMAT(1H ,20A4)
7     FORMAT(5X,'*** PSD CALCULATIONS ***'/
     1' WHAT IS THE FILE NAME?'/' THE FORMAT SHOULD',
     2' BE "VOL:NAME" WITH EXACTLY 12 CHARACTERS.')
8     FORMAT(I2,F10.4)
9     FORMAT(' HARMONIC PURITY IS',F10.4)
10    FORMAT(F10.4)
11    FORMAT(' POINT-TO-POINT SPACING =',F9.4,' um')
12    FORMAT(' POINT SPACING OF PSD =',F10.4,' INVERSE um')
13    FORMAT(//' PROFILE',I3)
14    FORMAT(5E12.4)
15    FORMAT(A12)
16    FORMAT(' FREQUENCY RANGE IS FROM',F10.4, ' TO',F10.4,
     1' INVERSE um')
17    FORMAT(16I5)
18    FORMAT(I2)
19    FORMAT(' I CAN"T READ THIS NAME. TRY AGAIN.')
20    FORMAT(/' AVERAGE PSD FOR ALL ',I2,' POSITIONS')
21    FORMAT(' RMS ROUGHNESS CALCULATED FROM THE INTEGRAL ',
     1'OF THE PSD = ',F10.5,' um')
23    FORMAT(' I CAN"T OPEN THIS FILE'/
     1' TYPE THE FILE NAME OVER AND GET IT RIGHT THIS TIME.')
24    FORMAT(' PSD (um)**3')
26    FORMAT(' AVE WAVELENGTH CALCULATED FROM PSD IS',F10.4,' um')
27    FORMAT(1H ,A16)
28    FORMAT(1H1,5X,' ***** PSD CALCULATIONS *****'///)
29    FORMAT(//' DO YOU WANT A PRINTOUT OF THE '/
     1' PSD NUMBERS? "Y" OR "N"?')
```

```
31    FORMAT(' THE FIRST AND LAST PROFILE POINTS ARE',I5,' & ',I5)
33    FORMAT(' BEGIN FFT')
34    FORMAT(' END FFT')
C
C
C FIRST, THE PROGRAM READS THE NAME OF THE DATA FILE TO BE
C ANALYZED FROM LU-5. THE NAME OF THE DATA FILE MUST HAVE
C THE FORMAT "VOL:ADATE", WHERE "VOL" IS THE 3 CHARACTER
C VOLUME NAME AND "ADATE" IS THE 8 CHARACTER FILE NAME.
C THE DATA FILE IS ASSIGNED LU-10. THEN THE PROGRAM
C READS THE SPECIMEN INFO, THE VERTICAL KCAL, AND THE
C PROFILE DATA FROM THE FILE "VOL:ADATE.DAT".
C
C
      ITRY = 0
92    WRITE(6,7)
      READ(5,15,ERR=93,END=502)DATFIL(1:12)
      GO TO 94
93    ITRY = ITRY + 1
      IF (ITRY .GT. 2) GO TO 502
      WRITE(6,19)
      GO TO 92
94    OPEN(UNIT=10,ERR=91,FILE=DATFIL(1:16),STATUS='OLD',
     1ACCESS='DIRECT',FORM='FORMATTED',RECL=80,BLANK='ZERO')
      GO TO 90
91    WRITE(6,23)
      ITRY = ITRY + 1
      IF (ITRY .GT. 2) GO TO 502
      GO TO 92
C
C
90    N = 2000
      RN = FLOAT(N)
      K = 500
      KPLUS = K+1
      IAVRG = 0
C
      CALL EZFFTI(N,WSAVE)
C
      READ(10,3,REC=2)SPEC
      READ(10,10,REC=4) DELTAX
      READ(10,4,REC=6)KCAL
      WRITE(3,28)
      WRITE(3,27) DATFIL(1:16)
      WRITE(3,6) SPEC
      WRITE (6,11) DELTAX
      WRITE(3,11) DELTAX
      WRITE(3,5) KCAL
      WRITE(6,29)
      READ(5,2) QUERY3
C
      DELTAF = 2./(RN*DELTAX)
      WRITE(3,12) DELTAF
        DO 113 I=1,K-1
        F(I) = I*DELTAF
        AVPSD(I) = 0.
 113    CONTINUE
C
```

```
      CALL PLOTS(0,0,0)
      CALL FACTOR(.7874)
      CALL PLOT(.5,.5,-3)
      READ(10,18,REC=7) NTOT
C
C
C NOW, THE OPERATOR IS ASKED TO CHOOSE WHETHER ALL
C THE PROFILES ARE TO BE ANALYZED AND, IF NOT, HOW MANY.
C
C
      CALL CHOICE(NTOT,ITOPRO,PRONUM)
C
C
C THE MAIN LOOP TO CALCULATE AND PLOT THE PSD FOR EACH
C PROFILE BEGINS HERE
C
C
      RTOPRO = ITOPRO
      DO 300 M = 1,ITOPRO
        RNUMBR = PRONUM(M)
        START = (PRONUM(M) - 1)*251 + 8
        READ (10,8,  REC=START) NRUN,RA
        WRITE(3,13) NRUN
        DO 301 M1=1,2
          READ (10,17) PROFIL
          WRITE(6,31) PROFIL(1),PROFIL(2000)
C
          CALL LEASQ(PROFIL,2000)
C
C
          WRITE(6,33)
          DO 310 I=1,N
            WSAVE(I) = KCAL*PROFIL(I)
310       CONTINUE
          IF (M1 .EQ. 1) CALL EZPSDF(DELTAX,N,DCLEV1,PSD,WSAVE)
          IF (M1 .EQ. 2) CALL EZPSDF(DELTAX,N,DCLEV2,PSD2,WSAVE)
          WRITE(6,34)
301     CONTINUE
        SQCHK = 0.
        FCHK = 0.
        DO 110 L=1,K
          PSD(L) = (PSD(L) +   PSD2(L))/2.
          FCHK = FCHK + F(L)*PSD(L)
          SQCHK = SQCHK + PSD(L)
          PSDLOG(L) = ALOG10(PSD(L))
          AVPSD(L) = AVPSD(L) + PSD(L)/RTOPRO
110     CONTINUE
       RMSCHK = SQRT(SQCHK*DELTAF)
       LAMBDA = SQCHK/FCHK
       FUNDAM = PSD(16) + PSD(17) + PSD(18) + PSD(19) + PSD(20)
       PURITY = FUNDAM/SQCHK
       DCLEVL = (DCLEV1 + DCLEV2)/2.
C
C
C NOW WE WRITE ALL THE RESULTS ON THE PRINTER.
C
C
       WRITE(3,1) DCLEVL
```

```
      WRITE(3,21) RMSCHK
      WRITE(3,9) PURITY
      WRITE(3,26)LAMBDA
      IF (QUERY3 .EQ. 'N') GO TO 202
      WRITE(3,24)
      WRITE(3,14) (PSD(I),I=1,K)
202   WRITE(3,16)  F(1),F(K-1)
C
C
C THE PLOT  ROUTINES ARE NOW CALLED AND THE LOG OF THE
C PSD IS PLOTTED.
C
C
      CALL PLOTME(F,PSDLOG,KPLUS,
     1'LOG OF [PSD (um CUBED)]',23,
     2'FREQUENCY (INVERSE um)',22,SPEC,RMSCHK,NRUN,IAVRG)
300   CONTINUE
C
C
C THE MAIN LOOP IS COMPLETE. NOW, WE PLOT AND PRINT THE
C RESULTS FOR THE AVERAGE  PSD OF ALL  THE  SETS IN  THE
C FILE.
C
C
      IAVRG = 1
      AVSQCK  = 0.
      AVFCHK = 0.
        DO  302 L=1,K
        AVSQCK = AVSQCK+AVPSD(L)
        AVFCHK = AVFCHK+F(L)*AVPSD(L)
        PSDLOG(L) = ALOG10(AVPSD(L))
302     CONTINUE
      AVRMS = SQRT(AVSQCK*DELTAF)
      AVLMDA  = AVSQCK/AVFCHK
      WRITE(3,20)  ITOPRO
      WRITE(3,21)AVRMS
      WRITE(3,26) AVLMDA
      WRITE(3,24)
      WRITE(3,14) (AVPSD(I),I=1,K)
      WRITE(3,16) F(1),F(K-1)
      CALL  PLOTME(F,PSDLOG,KPLUS,
     1'LOG OF [PSD  (um CUBED)]',23,
     2'FREQUENCY (INVERSE um)',22,SPEC,AVRMS,ITOPRO,IAVRG)
      CALL  PLOT(0.,0.,999)
502   STOP
      END
C
C
C
      SUBROUTINE PLOTME(X,Y,KPLUS,YLABEL,NYCHAR,XLABEL,
     1 NXCHAR,SPEC,RMS,NRUN,IAVRG)
      DIMENSION  X(KPLUS),Y(KPLUS),XLABEL(20),YLABEL(20),
     1 SPEC(15)
      RUN  = NRUN
      K = KPLUS-2
      CALL SCALE(Y,8.,K,+1)
      CALL SCALE(X,12.,K,+1)
      CALL NEWPEN(-1)
```

```
          Y(K+2) = -Y(K+2)
          Y(K+1) = Y(K+1)  -  8.*Y(K+2)
          CALL LINE(Y,X,K,1,0,31)
          CALL AXIS(0.,0.,YLABEL,-NYCHAR,9.,0.,Y(K+1),Y(K+2))
          CALL AXIS(9.,0.,XLABEL,-NXCHAR,12.,90.,X(K+1),X(K+2))
          CALL SYMBOL(0.,.2,.2,SPEC,90.,60)
          CALL SYMBOL(.5,.2,.2,'RMS ROUGHNESS = ',90.,16)
          IF (RMS .LT. 0.01) GO TO 501
          CALL NUMBER(.5,3.4,.2,RMS,90.,+3)
          CALL SYMBOL ( 0.4,4.6,0.2,7,90.,-1)
          CALL SYMBOL(.4,4.8,.2,109,90.,-1)
          GO TO 502
  501     CALL NUMBER (0.5,3.4,0.2,RMS,90.,+5)
          CALL SYMBOL (0.4,5.0,0.2,7,90.,-1)
          CALL SYMBOL (0.4,5.2,0.2,109,90.,-1)
  502     IF (IAVRG.EQ.1) GO TO 400
          CALL SYMBOL(1.,.2,.2,'PSD FOR POSITION',90.,16)
          CALL NUMBER(1.,3.7,.2,RUN,90.,-1)
          GO TO 401
  400     CALL SYMBOL(1.,.2,.2,'AVERAGE PSD FOR ALL ',90.,20)
          CALL NUMBER(1.,4.3,.2,RUN,90.,-1)
          CALL SYMBOL(1.,4.7,.2,'POSITIONS',90.,10)
  401     CALL PLOT(0.,0.,+23)
          RETURN
          END
  C
  C
  C THIS ROUTINE CALCULATES THE LEAST SQUARES STRAIGHT LINE FOR
  C A SET OF 2000 EQUALLY SPACED DATA POINTS. IT CAN BE USED TO
  C FILTER OUT ANY SLOPE IN A SET OF PROFILE DATA.
  C
  C
        SUBROUTINE LEASQ(PROFIL,N)
        INTEGER*2 PROFIL(N)
  1     FORMAT(' AZERO',11X,'AONE'/2E15.7)
        RN = N
  C
  C IN THE LEAST SQUARES FIT, WE HAVE ALREADY
  C CALCULATED THE SUM OF THE X'S AND X SQUARES,
  C WHOSE VALUES NEVER CHANGE. THIS AVOIDS HAVING
  C TO CALCULATE THEIR VALUES IN THE LOOP.
  C
        SUMX1 = 2001000.
        SUMX2 = 2.668667E9
        SUMY1 = 0.
        SUMXY = 0.
        DO 101 I=1,N
          SUMY1 = SUMY1 + PROFIL(I)
          SUMXY = SUMXY + I*PROFIL(I)
  101   CONTINUE
  C
        DELTA = RN*SUMX2 - SUMX1*SUMX1
        AZERO = (SUMX2*SUMY1 - SUMX1*SUMXY)/DELTA
        AONE = (RN*SUMXY - SUMX1*SUMY1)/DELTA
        WRITE(3,1) AZERO,AONE
  C
        DO 102 I=1,N
          RI = I
```

```
            SUB = AZERO + AONE*RI
            X = 0.5
            IF (SUB .LT. 0.) X = -X
            NSUB = SUB + X
            PROFIL(I) = PROFIL(I) - NSUB
102     CONTINUE
        RETURN
        END
C
C
C SUBROUTINE FOR DETERMINING WHICH PROFILES IN
C THE FILE ARE ANALYZED
C
C
        SUBROUTINE CHOICE(NTOT,ITOPRO,PRONUM)
        INTEGER*2 QUERY2,PRONUM(50)
C
1       FORMAT(' THERE ARE',I3,' PROFILES'//' DO YOU',
       1' WANT TO ANALYZE ALL OF THEM?'/' "Y" OR "N"?')
2       FORMAT(I2)
3       FORMAT(//' HOW MANY PROFILES DO YOU WANT TO ANALYZE?'/
       1' USE I2 FORMAT.')
4       FORMAT(//' TYPE THE NUMBER OF THE FIRST PROFILE TO BE ANALYZED.'/
       1' USE I2 FORMAT.')
5       FORMAT(A2)
6       FORMAT(//' TYPE THE NUMBER OF THE NEXT PROFILE TO BE ANALYZED.'/
       1' USE I2 FORMAT.')
C
        WRITE (6,1) NTOT
        READ (5,5) QUERY2
        IF ( QUERY2 .EQ. 'N ') GO TO 131
C
        ITOPRO = NTOT
        DO 132 I = 1,ITOPRO
132     PRONUM(I) = I
        RETURN
C
131     WRITE (6,3)
        READ (5,2) ITOPRO
        WRITE (6,4)
        READ (5,2) PRONUM(1)
        DO 133 I=2,ITOPRO
        WRITE (6,6)
133     READ (5,2) PRONUM(I)
        RETURN
        END
C
C
C
C
        SUBROUTINE RFFTF (N,R,WSAVE)
        DIMENSION       R(1)       ,WSAVE(1)
        CALL RFFTF1 (N,R,WSAVE,WSAVE(N+1),WSAVE(2*N+1))
        RETURN
        END
C
C
C
```

```
C
C
      SUBROUTINE RADF5 (IDO,L1,CC,CH,WA1,WA2,WA3,WA4)
      DIMENSION       CC(IDO,L1,5)           ,CH(IDO,5,L1)           ,
     1                WA1(1)     ,WA2(1)     ,WA3(1)     ,WA4(1)
      DATA TR11,TI11,TR12,TI12 /.309016994374947,.951056516295154,
     1-.809016994374947,.587785252292473/
      DO 101 K=1,L1
         CR2 = CC(1,K,5)+CC(1,K,2)
         CI5 = CC(1,K,5)-CC(1,K,2)
         CR3 = CC(1,K,4)+CC(1,K,3)
         CI4 = CC(1,K,4)-CC(1,K,3)
         CH(1,1,K) = CC(1,K,1)+CR2+CR3
         CH(IDO,2,K) = CC(1,K,1)+TR11*CR2+TR12*CR3
         CH(1,3,K) = TI11*CI5+TI12*CI4
         CH(IDO,4,K) = CC(1,K,1)+TR12*CR2+TR11*CR3
         CH(1,5,K) = TI12*CI5-TI11*CI4
  101 CONTINUE
      IF (IDO .EQ. 1) RETURN
      IDP2 = IDO+2
      DO 103 K=1,L1
         DO 102 I=3,IDO,2
            IC = IDP2-I
            DR2 = WA1(I-2)*CC(I-1,K,2)+WA1(I-1)*CC(I,K,2)
            DI2 = WA1(I-2)*CC(I,K,2)-WA1(I-1)*CC(I-1,K,2)
            DR3 = WA2(I-2)*CC(I-1,K,3)+WA2(I-1)*CC(I,K,3)
            DI3 = WA2(I-2)*CC(I,K,3)-WA2(I-1)*CC(I-1,K,3)
            DR4 = WA3(I-2)*CC(I-1,K,4)+WA3(I-1)*CC(I,K,4)
            DI4 = WA3(I-2)*CC(I,K,4)-WA3(I-1)*CC(I-1,K,4)
            DR5 = WA4(I-2)*CC(I-1,K,5)+WA4(I-1)*CC(I,K,5)
            DI5 = WA4(I-2)*CC(I,K,5)-WA4(I-1)*CC(I-1,K,5)
            CR2 = DR2+DR5
            CI5 = DR5-DR2
            CR5 = DI2-DI5
            CI2 = DI2+DI5
            CR3 = DR3+DR4
            CI4 = DR4-DR3
            CR4 = DI3-DI4
            CI3 = DI3+DI4
            CH(I-1,1,K) = CC(I-1,K,1)+CR2+CR3
            CH(I,1,K) = CC(I,K,1)+CI2+CI3
            TR2 = CC(I-1,K,1)+TR11*CR2+TR12*CR3
            TI2 = CC(I,K,1)+TR11*CI2+TR12*CI3
            TR3 = CC(I-1,K,1)+TR12*CR2+TR11*CR3
            TI3 = CC(I,K,1)+TR12*CI2+TR11*CI3
            TR5 = TI11*CR5+TI12*CR4
            TI5 = TI11*CI5+TI12*CI4
            TR4 = TI12*CR5-TI11*CR4
            TI4 = TI12*CI5-TI11*CI4
            CH(I-1,3,K) = TR2+TR5
            CH(IC-1,2,K) = TR2-TR5
            CH(I,3,K) = TI2+TI5
            CH(IC,2,K) = TI5-TI2
            CH(I-1,5,K) = TR3+TR4
            CH(IC-1,4,K) = TR3-TR4
            CH(I,5,K) = TI3+TI4
            CH(IC,4,K) = TI4-TI3
  102    CONTINUE
```

```
  103 CONTINUE
      RETURN
      END
C
C
C
C
C
      SUBROUTINE EZFFT1 (N,WA,IFAC)
      DIMENSION       WA(1)      ,IFAC(1)    ,NTRYH(4)
      DATA NTRYH(1),NTRYH(2),NTRYH(3),NTRYH(4)/4,2,3,5/
     1    ,TPI/6.28318530717959/
      NL = N
      NF = 0
      J = 0
  101 J = J+1
      IF (J-4) 102,102,103
  102 NTRY = NTRYH(J)
      GO TO 104
  103 NTRY = NTRY+2
  104 NQ = NL/NTRY
      NR = NL-NTRY*NQ
      IF (NR) 101,105,101
  105 NF = NF+1
      IFAC(NF+2) = NTRY
      NL = NQ
      IF (NTRY .NE. 2) GO TO 107
      IF (NF .EQ. 1) GO TO 107
      DO 106 I=2,NF
         IB = NF-I+2
         IFAC(IB+2) = IFAC(IB+1)
  106 CONTINUE
      IFAC(3) = 2
  107 IF (NL .NE. 1) GO TO 104
      IFAC(1) = N
      IFAC(2) = NF
      ARGH = TPI/FLOAT(N)
      IS = 0
      NFM1 = NF-1
      L1 = 1
      IF (NFM1 .EQ. 0) RETURN
      DO 111 K1=1,NFM1
         IP = IFAC(K1+2)
         L2 = L1*IP
         IDO = N/L2
         IPM = IP-1
         ARG1 = FLOAT(L1)*ARGH
         CH1 = 1.
         SH1 = 0.
         DCH1 = COS(ARG1)
         DSH1 = SIN(ARG1)
         DO 110 J=1,IPM
            CH1H = DCH1*CH1-DSH1*SH1
            SH1 = DCH1*SH1+DSH1*CH1
            CH1 = CH1H
            I = IS+2
            WA(I-1) = CH1
            WA(I) = SH1
```

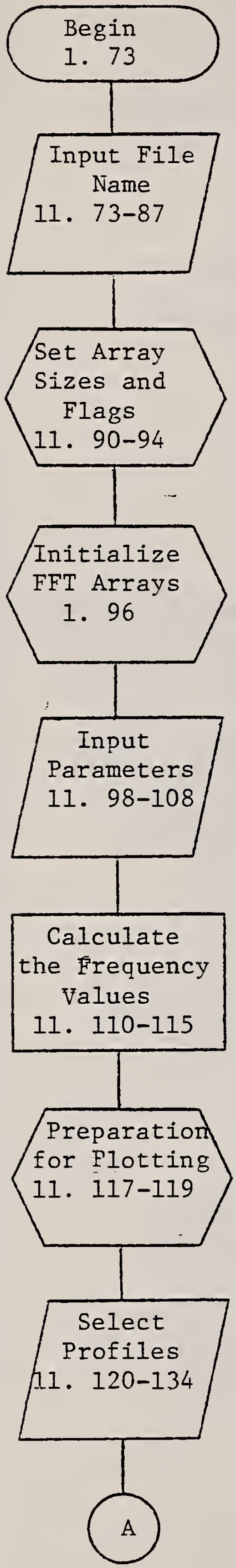
```
                IF (IDO .LT. 5) GO TO 109
                DO 108 II=5,IDO,2
                   I = I+2
                   WA(I-1) = CH1*WA(I-3)-SH1*WA(I-2)
                   WA(I) = CH1*WA(I-2)+SH1*WA(I-3)
  108           CONTINUE
  109           IS = IS+IDO
  110        CONTINUE
             L1 = L2
  111 CONTINUE
      RETURN
      END
C
C
C
C
C
      SUBROUTINE EZFFTI (N,WSAVE)
      DIMENSION       WSAVE(1)
      CALL EZFFT1 (N,WSAVE(2*N+1),WSAVE(3*N+1))
      RETURN
      END
C
C
C
C
C
C THE FOLLOWING SUBROUTINE IS MODIFIED FROM THE
C EZFFTF SUBROUTINE SUPPLIED TO US. THIS ONE CAL-
C CULATES THE POWER DIRECTLY, NOT JUST THE FOURIER
C AMPLITUDES.
C
      SUBROUTINE EZPSDF (DELTAX,N,DCLEVL,PSD,WSAVE)
C
C                     VERSION 3  JUNE 1979
C
      DIMENSION       PSD(1)       ,WSAVE(1)
C
      CALL RFFTF (N,WSAVE,WSAVE(N+1))
      CF = 1./FLOAT(N)
      CFM = -CF
      DCLEVL = CF*WSAVE(1)
      NS2 = (N+1)/2
      NS2M = NS2-1
      DO 105 I=2,NS2M,2
         REAL = WSAVE(2*I)
         RIMAG = WSAVE(2*I+1)
         DNREAL = WSAVE(2*I-2)
         DNIMAG = WSAVE(2*I-1)
         UPREAL = WSAVE(2*I+2)
         UPIMAG = WSAVE(2*I+3)
         RMID = REAL*REAL + RIMAG*RIMAG
         RDN = DNREAL*DNREAL + DNIMAG*DNIMAG
         RUP = UPREAL*UPREAL + UPIMAG*UPIMAG
C
C THE FACTOR OF 2 IS NEEDED BECAUSE WE ARE COMPUTING A
C ONE-SIDED PSD. WHEN WE INTEGRATE TO GET POWER, WE SHOULD
C INCLUDE THE NEGATIVE FREQUENCY CONTRIBUTION. INSTEAD,
```

```
C WE MULTIPLY THE POSITIVE SIDE BY A FACTOR OF 2.
C
C
C IN ADDITION, INSTEAD OF CALCULATING A 1000 POINT POWER
C SPECTRUM FROM THE 2000 POINT PROFILE, WE CALULATE A SMOOTHED
C 500 POINT POWER SPECTRUM.
C
C
          PSD(I/2) = 2.*DELTAX*CF*(.25*RDN + .5*RMID + .25*RUP)
  105 CONTINUE
      PSD(NS2/2) = 2.*DELTAX*CF*WSAVE(N)*WSAVE(N)
      RETURN
      END
C
C
C
C
C
      SUBROUTINE RADF2 (IDO,L1,CC,CH,WA1)
      DIMENSION       CH(IDO,2,L1)           ,CC(IDO,L1,2)           ,
     1                WA1(1)
      DO 101 K=1,L1
         CH(1,1,K) = CC(1,K,1)+CC(1,K,2)
         CH(IDO,2,K) = CC(1,K,1)-CC(1,K,2)
  101 CONTINUE
      IF (IDO-2) 107,105,102
  102 IDP2 = IDO+2
      DO 104 K=1,L1
         DO 103 I=3,IDO,2
            IC = IDP2-I
            TR2 = WA1(I-2)*CC(I-1,K,2)+WA1(I-1)*CC(I,K,2)
            TI2 = WA1(I-2)*CC(I,K,2)-WA1(I-1)*CC(I-1,K,2)
            CH(I,1,K) = CC(I,K,1)+TI2
            CH(IC,2,K) = TI2-CC(I,K,1)
            CH(I-1,1,K) = CC(I-1,K,1)+TR2
            CH(IC-1,2,K) = CC(I-1,K,1)-TR2
  103    CONTINUE
  104 CONTINUE
      IF (MOD(IDO,2) .EQ. 1) RETURN
  105 DO 106 K=1,L1
         CH(1,2,K) = -CC(IDO,K,2)
         CH(IDO,1,K) = CC(IDO,K,1)
  106 CONTINUE
  107 RETURN
      END
C
C
C
C
C
      SUBROUTINE RADF4 (IDO,L1,CC,CH,WA1,WA2,WA3)
      DIMENSION       CC(IDO,L1,4)           ,CH(IDO,4,L1)           ,
     1                WA1(1)     ,WA2(1)     ,WA3(1)
      DATA HSQT2 /.7071067811865475/
      DO 101 K=1,L1
         TR1 = CC(1,K,2)+CC(1,K,4)
         TR2 = CC(1,K,1)+CC(1,K,3)
         CH(1,1,K) = TR1+TR2
```

```
            CH(IDO,4,K) = TR2-TR1
            CH(IDO,2,K) = CC(1,K,1)-CC(1,K,3)
            CH(1,3,K) = CC(1,K,4)-CC(1,K,2)
  101    CONTINUE
         IF (IDO-2) 107,105,102
  102    IDP2 = IDO+2
         DO 104 K=1,L1
            DO 103 I=3,IDO,2
               IC = IDP2-I
               CR2 = WA1(I-2)*CC(I-1,K,2)+WA1(I-1)*CC(I,K,2)
               CI2 = WA1(I-2)*CC(I,K,2)-WA1(I-1)*CC(I-1,K,2)
               CR3 = WA2(I-2)*CC(I-1,K,3)+WA2(I-1)*CC(I,K,3)
               CI3 = WA2(I-2)*CC(I,K,3)-WA2(I-1)*CC(I-1,K,3)
               CR4 = WA3(I-2)*CC(I-1,K,4)+WA3(I-1)*CC(I,K,4)
               CI4 = WA3(I-2)*CC(I,K,4)-WA3(I-1)*CC(I-1,K,4)
               TR1 = CR2+CR4
               TR4 = CR4-CR2
               TI1 = CI2+CI4
               TI4 = CI2-CI4
               TI2 = CC(I,K,1)+CI3
               TI3 = CC(I,K,1)-CI3
               TR2 = CC(I-1,K,1)+CR3
               TR3 = CC(I-1,K,1)-CR3
               CH(I-1,1,K) = TR1+TR2
               CH(IC-1,4,K) = TR2-TR1
               CH(I,1,K) = TI1+TI2
               CH(IC,4,K) = TI1-TI2
               CH(I-1,3,K) = TI4+TR3
               CH(IC-1,2,K) = TR3-TI4
               CH(I,3,K) = TR4+TI3
               CH(IC,2,K) = TR4-TI3
  103       CONTINUE
  104    CONTINUE
         IF (MOD(IDO,2) .EQ. 1) RETURN
  105    CONTINUE
         DO 106 K=1,L1
            TI1 = -HSQT2*(CC(IDO,K,2)+CC(IDO,K,4))
            TR1 = HSQT2*(CC(IDO,K,2)-CC(IDO,K,4))
            CH(IDO,1,K) = TR1+CC(IDO,K,1)
            CH(IDO,3,K) = CC(IDO,K,1)-TR1
            CH(1,2,K) = TI1-CC(IDO,K,3)
            CH(1,4,K) = TI1+CC(IDO,K,3)
  106    CONTINUE
  107    RETURN
         END
C
C
C
C
C
         SUBROUTINE RFFTF1 (N,C,CH,WA,IFAC)
         DIMENSION       CH(1)      ,C(1)       ,WA(1)      ,IFAC(1)
         NF = IFAC(2)
         NA = 1
         L2 = N
         IW = N
         DO 111 K1=1,NF
            KH = NF-K1
```

```
         IP = IFAC(KH+3)
         L1 = L2/IP
         IDO = N/L2
         IDL1 = IDO*L1
         IW = IW-(IP-1)*IDO
         NA = 1-NA
         IF (IP .NE. 4) GO TO 102
         IX2 = IW+IDO
         IX3 = IX2+IDO
         IF (NA .NE. 0) GO TO 101
         CALL RADF4 (IDO,L1,C,CH,WA(IW),WA(IX2),WA(IX3))
         GO TO 110
  101    CALL RADF4 (IDO,L1,CH,C,WA(IW),WA(IX2),WA(IX3))
         GO TO 110
  102    IF (IP .NE. 2) GO TO 104
         IF (NA .NE. 0) GO TO 103
         CALL RADF2 (IDO,L1,C,CH,WA(IW))
         GO TO 110
  103    CALL RADF2 (IDO,L1,CH,C,WA(IW))
         GO TO 110
 104     IX2 = IW+IDO
         IX3 = IX2+IDO
         IX4 = IX3+IDO
         IF (NA .NE. 0) GO TO 107
         CALL RADF5 (IDO,L1,C,CH,WA(IW),WA(IX2),WA(IX3),WA(IX4))
         GO TO 110
  107    CALL RADF5 (IDO,L1,CH,C,WA(IW),WA(IX2),WA(IX3),WA(IX4))
  110    L2 = L1
  111 CONTINUE
      IF (NA .EQ. 1) RETURN
      DO 112 I=1,N
         C(I) = CH(I)
  112 CONTINUE
      RETURN
      END
```

11.4 Flowchart for PSD



The operator is prompted to type the name of the data file to be examined. The file is then opened as a logical unit.

There are 2000 points to be analyzed for each calculation of a 500 point PSD.

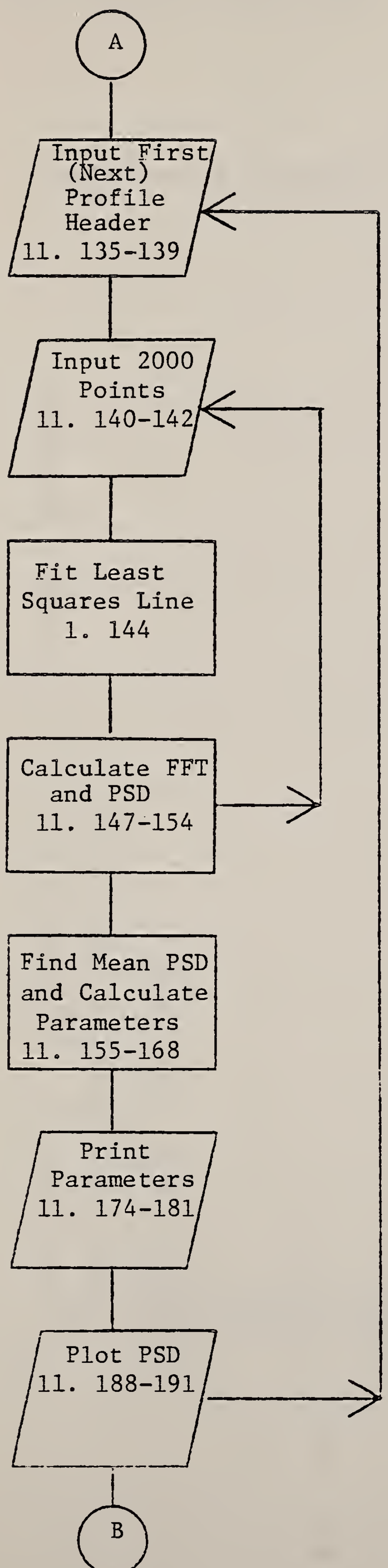
The main program calls EZFFTI, which in turn calls EZFFT1.

The program reads the specimen I.D., the horizontal spacing of the data points, and the instrument calibration constant and prompts the operator to choose whether or not to print the values of the PSD.

These frequency values comprise the independent variable of the PSD.

Three subroutines are called to initialize the plotting sequence.

The total number of profiles in the data file is read. Then the operator is asked to choose which of these will be analyzed. The subroutine CHOICE is used for this task.



The main loop to calculate and plot the PSD begins.

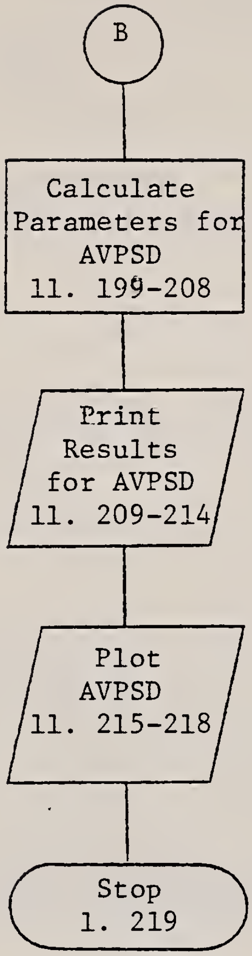
The inner loop to calculate the PSD for each half of the profile begins.

The subroutine EZPSDF is called and in turn calls several others.

The mean PSD for the two halves is averaged in this section. The PSD is used to calculate values for R_q and a mean wavelength parameter.

The PSD values are also printed if called for.

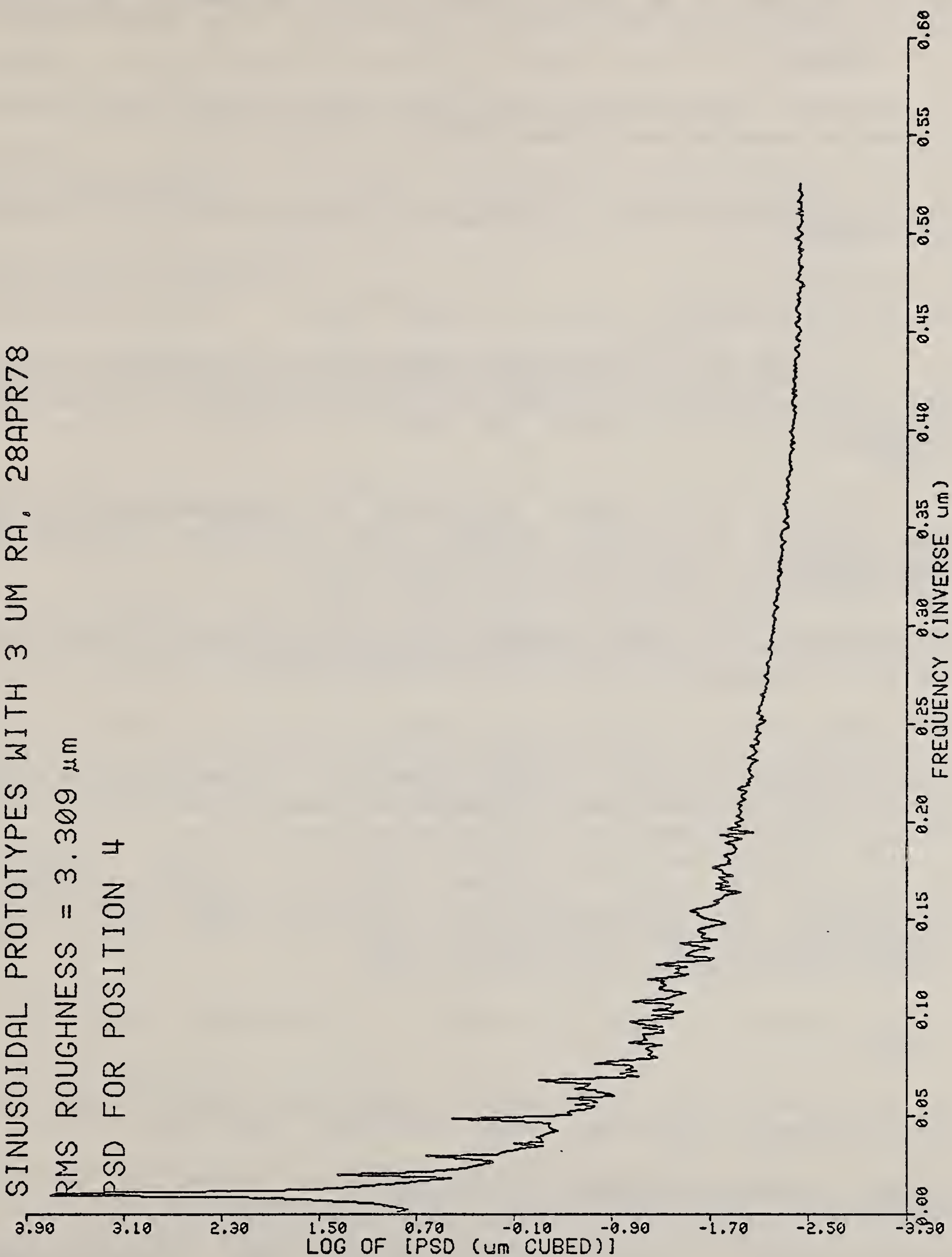
PLOTME is called here.



R_q and the mean wavelength parameter are calculated again.

PLOTME is called again.

11.5 Example of PSD Plot



12. References

1. D. Byrne, P. A. Fitzsimmons, A. K. Brook, Paper E-13 in Proc. Symposium on Ship Costs and Energy '82 (Society of Naval Architects and Marine Engineers, New York, To be published).

2. B. Scheffer, Comparison de Differentes Normes Nationales (Régie Nationale des Usines Renault, Paris, 1969).

3. E. L. Church, M. R. Howells, and T. V. Vorburger in Reflecting Optics for Synchrotron Radiation, Proc. SPIE 315, 202 (1981).

4. T. V. Vorburger, E. C. Teague, and F. E. Scire, Dimensions/NBS 62, No. 11, 18 (Nov. 1978).

5. E. C. Teague, Evaluation, Revision, and Application of the NBS Stylus/Computer System for the Measurement of Surface Roughness, NBS Tech. Note 902 (Dept. of Commerce, Wash., DC, 1976).

6. E. C. Teague, Evaluation, Revision, and Application of the NBS Stylus Computer System for Surface Roughness Measurement: Minicomputer Software, NBS Internal Report 75-924 (Dept. of Commerce, Wash., DC, 1975).

7. M. G. Natrella, Experimental Statistics, NBS Handbook 91 (Dept. of Commerce, Wash., DC, 1963) p. 1-10.

8. Ref. 5, p. 15.

9. R. C. Spragg and D. J. Whitehouse, Proceedings of the Institution of Mechanical Engineers, 185, (47/71) 697 (1970-71).

10. T. V. Vorburger, F. E. Scire, and E. C. Teague, Surface Roughness Measurements of Circular Disks and Their Correlation with Hydrodynamic Drag, NBS Tech. Note 1151 (Dept. of Commerce, Wash., DC, 1982).

11. E. C. Teague, F. E. Scire, and T. V. Vorburger, Wear 83, 61 (1982).

12. G. M. Jenkins and D. G. Watts, Spectral Analysis and Its Applications (Holden-Day, San Francisco, 1968) pp. 171 ff.

13. Guide to Available Mathematical Software, Center for Applied Mathematics (Nat. Bur. Stds., Wash. DC, 1981).

14. J. S. Bendat and A. G. Piersol, Random Data: Analysis and Measurement Procedures (Wiley-Interscience, New York, 1971) p. 318.

NBS-114A (REV. 2-80)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NBSIR 83-2703	2. Performing Organ. Report No.	3. Publication Date July 1983

4. TITLE AND SUBTITLE

FASTMENU: A Set of FORTRAN Programs for Analyzing Surface Texture

5. AUTHOR(S)

Theodore V. Vorburger

6. PERFORMING ORGANIZATION *(If joint or other than NBS, see instructions)*

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)*

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)*

A set of FORTRAN programs for surface texture analysis is described. These programs were developed for use with a minicomputer that is interfaced to stylus type instruments. The programs 1) perform data acquisition from the stylus instruments, 2) store the data on magnetic disk, and 3) perform statistical analyses for parameters such as the roughness average R_a, rms roughness R_q, and for the autocorrelation function and amplitude density function.

12. KEY WORDS *(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)*

amplitude density function; autocorrelation function; calibration; digitization; metrology; minicomputer; roughness; statistics; stylus; surface metrology; surface profile; surface texture

13. AVAILABILITY

☒ Unlimited
☐ For Official Distribution. Do Not Release to NTIS
☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.
☒ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

123

15. Price

$13.00